HUBBLEMIND

Garment Worker Productivity Prediction Project | HubbleMind Report

NJI RUTH MBIKANG
mbikangruth@gmail.com

December 29, 2024

Contents

1. Introduction
2. Data Overview
3. Data Preparation
4. Exploratory Data Analysis (EDA)
5. Machine Learning Model Selection and Evaluation
6. Conclusion
7. References

Introduction

Hubblemind

- This project aims to build a machine learning model that predicts productivity levels of garment workers based on various operational factors within a manufacturing setting.
- By analyzing features like work-in-progress, overtime, incentives, and team dynamics, we have developed models to predict continuous productivity values.
- This shall help businesses improve their workflow efficiency and resource management.
- With regards to this project we have worked on the dataset provided by Hubblemind having over 14 columns and 1,197 rows. Using python, we have performed some analysis on the dataset relating to Garment worker productivity and we intend providing insights that could benefit the industry.
- Code and new dataset can be accessed through this link: LINK

Data Overview

Hubblemind

The key features in the dataset includes:

- **date**: Date of the record.
- **quarter**: The quarter of the year (e.g., Q1, Q2).
- **department**: Department of workers (e.g., sewing, finishing).
- **team**: The number representing the team.
- **targeted_productivity**: The target productivity (between 0 and 1).
- **smv**: Standard Minute Value (time required to complete the task).
- **wip**: Work In Progress (missing values present).
- **over_time**: Overtime in minutes.
- **incentive**: Bonus paid to workers.
- **idle_time**: Time during which no work was done.
- **idle_men**: Number of idle workers.
- **no_of_style_change**: Number of style changes in production.
- **no_of_workers**: Number of workers in the team.
- **actual_productivity**: Target variable representing the productivity achieved (between 0 and 1)

Data Preparation

Hubblemind

Data Preparation

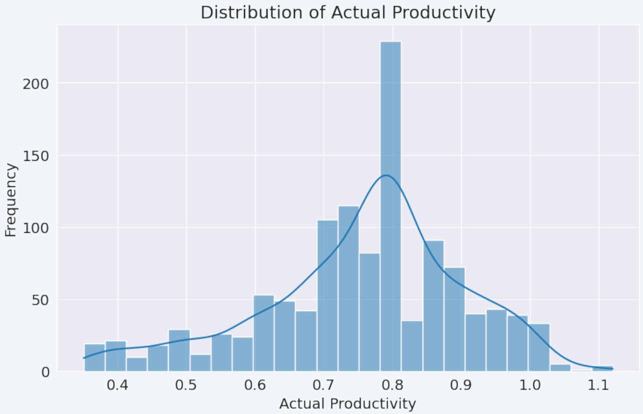
- The dataset used for this project is the **Garment Worker Productivity** Dataset. It includes 1,197 records and 14 features, covering different attributes related to the garment production process.
- We started off with uploading the dataset and viewing. It was seen that the **wip** variable had missing values. It was handled using the **median method**.
- This was followed by detecting and handling outliers in columns like idle_time, incentive, and actual_productivity.
- Next on, we identified numeric and categorical columns. The numeric columns were scaled to the $(0, 1)$ range. While we encoded categorical columns to one-hot vectors.
- Finally, we separated the target variable **actual_productivity** from the features which is every other variable.

Exploratory Data Analysis (EDA)

Hubblemind

Actual Productivity plot

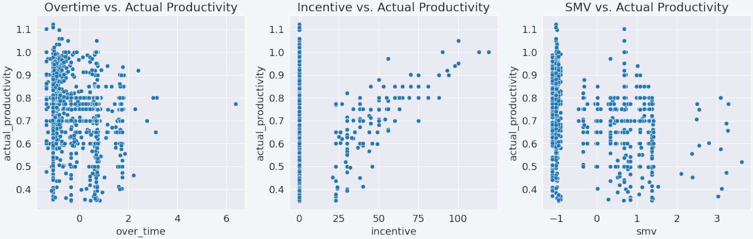
The figure shows the distribution of the actual productivity. It is seen that **0.8** has the highest frequency.



Distribution of the Actual Productivity

Scatter plot Comparison between between actual productivity and the incentive, smv and overtime

The figure shows the relationship between actual productivity and incentive, smv and overtime using a scatter plot.



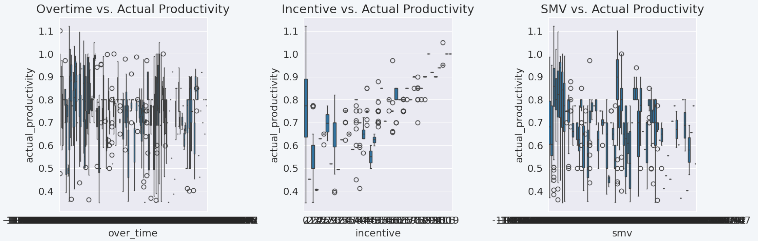
Scatter plot Comparison between between actual productivity and the incentive, smv and overtime.

Relationship Interpretation

- From the relationship between actual productivity and overtime, it shows that workers are more productive when the over time is limited to 2.
- For the second, the plot tells us that there is a team with productivity from 0 to 1.1 with zero incentives. However, the rest of the data shows a positive linear movement towards the increase of incentives.
- The last plot shows the relationship between the actual productivity and the standard minute value. We see that we have three clusters. The first is on smv = -1, the second is when smv equals 1 to 2 and the last is when smv is 2 to 3.

Box plot Comparison between actual productivity and the incentive, smv and overtime

The figure shows the relationship between actual productivity and incentive, smv and overtime in a box plot.

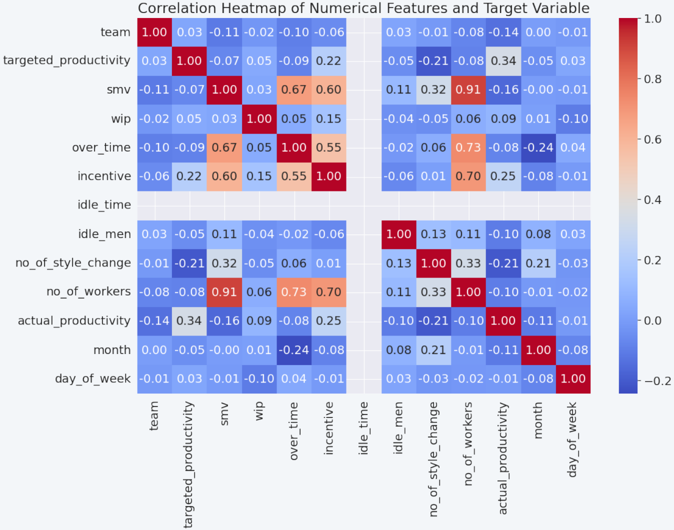


Box plot Comparison between actual productivity and incentive, smv and overtime.

Relationship Interpretation

- There is a similar pattern between the scatter plot and the box plot, hence the data is **robust**. There is a positive relationship between actual productivity and incentives.
- One could say the workers get more productive when given incentives.

Incentive has the highest correlation of **0.25** with the actual productivity. This is closely followed by no_of_style_change and smv with negative correlations of **-0.21 and -0.16**.



	team	targeted_productivity	smv	wip	over_time	incentive	idle_time	idle_men	no_of_style_change	no_of_workers	actual_productivity	month	day_of_week
team	1.00	0.03	-0.11	-0.02	-0.10	-0.06		0.03	-0.01	-0.08	-0.14	0.00	-0.01
targeted_productivity	0.03	1.00	-0.07	0.05	-0.09	0.22		-0.05	-0.21	-0.08	0.34	-0.05	0.03
smv	-0.11	-0.07	1.00	0.03	0.67	0.60		0.11	0.32	0.91	-0.16	-0.00	-0.01
wip	-0.02	0.05	0.03	1.00	0.05	0.15		-0.04	-0.05	0.06	0.09	0.01	-0.10
over_time	-0.10	-0.09	0.67	0.05	1.00	0.55		-0.02	0.06	0.73	-0.08	-0.24	0.04
incentive	-0.06	0.22	0.60	0.15	0.55	1.00		-0.06	0.01	0.70	0.25	-0.08	-0.01
idle_time													
idle_men	0.03	-0.05	0.11	-0.04	-0.02	-0.06		1.00	0.13	0.11	-0.10	0.08	0.03
no_of_style_change	-0.01	-0.21	0.32	-0.05	0.06	0.01		0.13	1.00	0.33	-0.21	0.21	-0.03
no_of_workers	-0.08	-0.08	0.91	0.06	0.73	0.70		0.11	0.33	1.00	-0.10	-0.01	-0.02
actual_productivity	-0.14	0.34	-0.16	0.09	-0.08	0.25		-0.10	-0.21	-0.10	1.00	-0.11	-0.01
month	0.00	-0.05	-0.00	0.01	-0.24	-0.08		0.08	0.21	-0.01	-0.11	1.00	-0.08
day_of_week	-0.01	0.03	-0.01	-0.10	0.04	-0.01		0.03	-0.03	-0.02	-0.01	-0.08	1.00

Correlation Heatmap of Numerical Features and Target Variable: actual_productivity

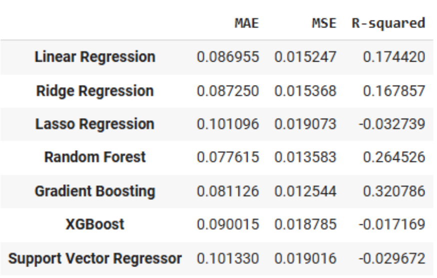
Machine Learning Model Selection and Evaluation

Hubblemind

Metric Explanation

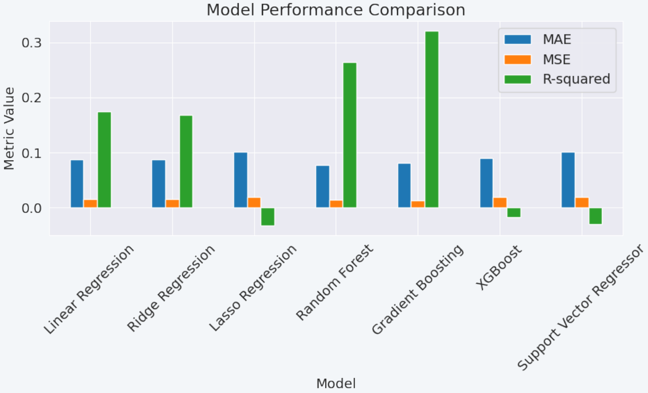
- **MAE (Mean Absolute Error)**: measures the average magnitude of the errors in a set of predictions, without considering their direction. It's the average of the absolute differences between the predicted and actual values. MAE ranges from 0 to infinity, with 0 meaning a perfect model.
- **MSE (Mean Squared Error**): measures the average of the squares of the errors. It's a way to quantify the difference between the model's predictions and the actual values. MSE ranges from 0 to infinity, with 0 meaning a perfect model.
- **R-squared (Coefficient of Determination)**: R-squared is a statistical measure that represents the proportion of the variance in the dependent variable (the target, like actual_productivity) that is predictable from the independent variables (the features, like over_time, incentive, etc.). It shows how well the model's predictions match the actual data. It ranges between 0 and 1. Higher R-squared values indicate a better fit.

Machine Learning Model Selection and Evaluation



	MAE	MSE	R-squared
Linear Regression	0.086955	0.015247	0.174420
Ridge Regression	0.087250	0.015368	0.167857
Lasso Regression	0.101096	0.019073	-0.032739
Random Forest	0.077615	0.013583	0.264526
Gradient Boosting	0.081126	0.012544	0.320786
XGBoost	0.090015	0.018785	-0.017169
Support Vector Regressor	0.101330	0.019016	-0.029672

Performance with target productivity as a feature

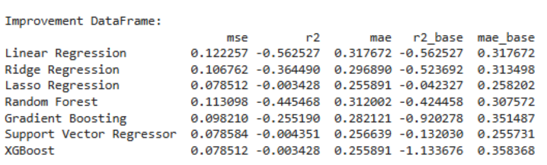


Model comparison with target productivity as a feature

Plot Interpretation

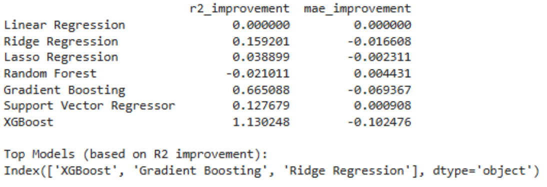
- Gradient boosting has least MSE and the highest R-squared when compared to the other models
- However, Target productivity being used as a feature is not good since it is similar to the actual productivity so it has to be taken out for us to train the models again. This is done in the subsequent slides.

Model Performances



```
Improvement DataFrame:
                              mse        r2       mae   r2_base  mae_base
Linear Regression        0.122257 -0.562527  0.317672 -0.562527  0.317672
Ridge Regression         0.106762 -0.364490  0.296890 -0.523692  0.313498
Lasso Regression         0.078512 -0.003428  0.255891 -0.042327  0.258202
Random Forest            0.113098 -0.445468  0.312002 -0.424458  0.307572
Gradient Boosting        0.098210 -0.255190  0.282121 -0.920278  0.351487
Support Vector Regressor 0.078584 -0.004351  0.256639 -0.132030  0.255731
XGBoost                  0.078512 -0.003428  0.255891 -1.133676  0.358368
```

Base Model and Hyperparameter tuning Performances



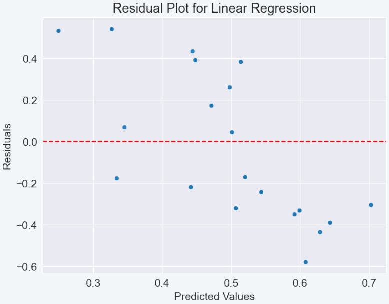
```
                          r2_improvement  mae_improvement
Linear Regression               0.000000         0.000000
Ridge Regression                0.159201        -0.016608
Lasso Regression                0.038899        -0.002311
Random Forest                  -0.021011         0.004431
Gradient Boosting               0.665088        -0.069367
Support Vector Regressor        0.127679         0.000908
XGBoost                         1.130248        -0.102476

Top Models (based on R2 improvement):
Index(['XGBoost', 'Gradient Boosting', 'Ridge Regression'], dtype='object')
```

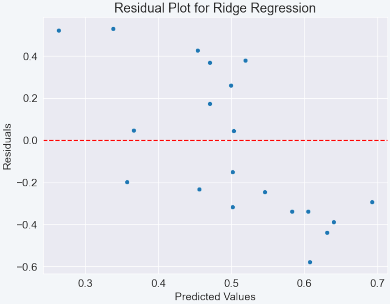
R2 improvement

We then trained the models without targeted productivity as a feature and got the following results for the base model and hyperparameter tuning for all 7 models. There is a significant improvement in hyperparameter tuning. The top three models based on the R2 improvement are XGboost, Gradient Boosting and Ridge Regression with values **1.13, 0.67 and 0.16** respectively.

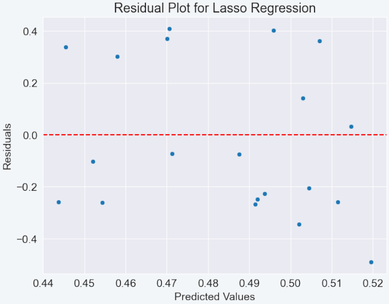
Residual Plots for Linear, Ridge and Lasso Regression



Linear Regression Residual Plot

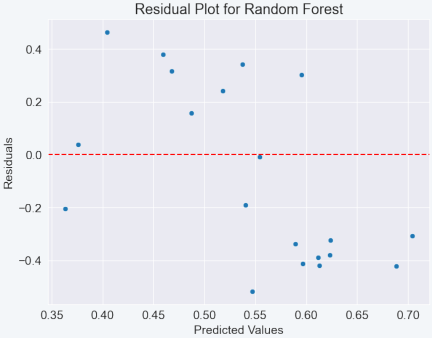


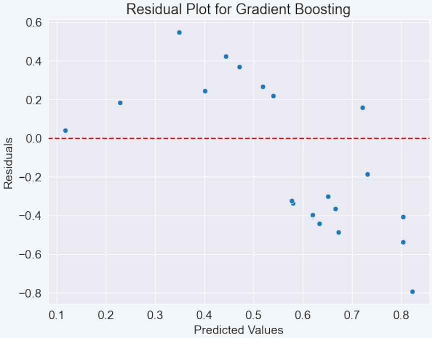
Ridge Regression Residual Plot



Lasso Regression Residual Plot

Residual Plots for Random Forest and Gradient Boosting

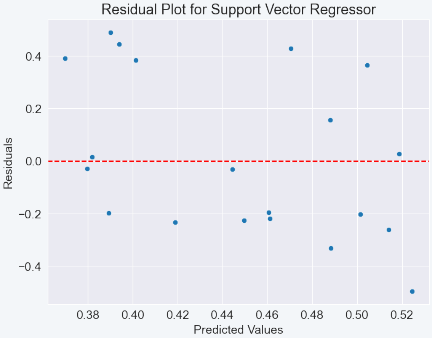




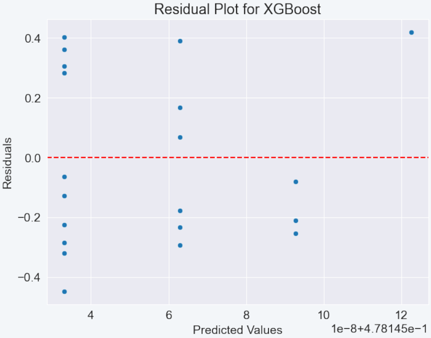
Residual Plot for Random Forest

Residual Plot Gradient Boosting

Residual Plots for Support vector Regressor and XGBoost



Residual Plot for Support vector Regressor



Residual Plot for XGBoost

Insights from the different Residual Plots

- **Linear Regression**: The plot shows a moderate scatter of residuals around the zero line which seems to be doing a reasonably good job, but there are some indications that it's not a perfect fit.
- **Ridge Regression**: The Ridge Regression model is performing almost identically to the Linear Regression model. The regularization penalty of Ridge didn't seem to make a significant difference in the fit for this our dataset.
- **Lasso Regression**: The Lasso Regression model is also performing almost identically to the Linear and Ridge Regression models. The L1 regularization of Lasso didn't seem to change the fit. The same subtle issues with linearity and inconsistent spread persist.

Insights from the different Residual Plots

- **Random Forest**: The Random Forest model is a much better fit for the data than the three linear models above. The lack of a curve and the more consistent spread suggest that it's capturing the non-linear relationships and has better even spread.
- **Gradient Boosting**: The Gradient Boosting model is a good fit for the data.
- **Support Vector Regressor**: The SVR model is an improvement over the linear models, but not quite as good as the tree-based models.
- **XGBoost**: The XGBoost model is an excellent fit for the data. It has effectively addressed the non-linearity issues that the linear models struggled with, and it maintains a good spread of residuals.

Conclusion

- The aim was to develop an ML-based predictive system to forecast garment worker productivity.
- The process involved data exploration, cleaning, feature engineering, model selection, and evaluation.
- We evaluated models including Linear, Ridge, Lasso, Random Forest, Gradient Boosting, Support Vector Regressor, and XGBoost.
- Linear models showed limited performance compared to tree-based models like Random Forest, Gradient Boosting, and XGBoost that performed better, with **XGBoost** emerging as the **best model** due to its superior accuracy, random residual distribution, and lowest error metrics.
- The final predictive system, built using the optimized XGBoost model, allows users to input operational data and receive a predicted productivity level. This tool can help improve decision-making and resource management in garment manufacturing.
- In conclusion, the project successfully built a robust predictive system, with potential for further improvement through additional data, feature refinement, and hyperparameter tuning.

References

References

- ”Introduction to Statistical Learning” by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani.
- ”Applied Regression Analysis” by Draper and Smith.
- https://www.statology.org/
- https://medium.com/