Productivity Prediction of Garment Employees

Final Report

Presented to the

Fusemachine Inc.

In Partial Fulfillment

of the requirements for the

H&M Presents Microdegree in AI

By

KAZI HASAN IBN ARIF
NAFIUR RAHMAN KHADEM

30 June 2022

Abstract

The apparel industry houses a huge amount and variety of data. At every step of the supply chain, data is collected and stored by each supply chain actor. This data, when used intelligently, can help with solving a good number of problems for the industry. The quality and quantity of the product produced will not be questioned each time it is created due to improved worker efficiency. So, it is highly desirable among the decision makers in the garment industry to track, analyze, and predict the productivity performance of the working teams in their factories. We developed a machine learning model that can predict whether a garment industry can achieve its target productivity or not based on some given parameters. Our best model can predict with a precision of 93%, a recall of 88%, and a weighted f1 score of 89%. We also analyzed the error of our model on the failure cases.

Keywords: EDA, Hyperparameter tuning, Pruning, Logistic Regression, Random Forest, AdaBoost, etc.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

Introduction to the Project

The garment industry is one of the key examples of the industrial globalisation of this modern era. It is a highly labour-intensive industry with lots of manual processes. Satisfying the huge global demand for garment products is mostly dependent on the production and delivery performance of the employees in the garment manufacturing companies. In many developing nations, such as Bangladesh, which is currently the second-largest exporter of apparel in the world behind China, the ready-to-wear garment industry plays a significant role in manufacturing production, employment, and trade (Chaerani, 2018). According to the Bureau of Export Promotion Data, which was recently published, ready-to-wear exports from Bangladesh generate roughly $ 30.61 billion in total export revenue, accounting for close to 14.07 percent of Bangladesh's GDP. The quality and quantity of the product produced will not be questioned each time it is created due to improved worker efficiency. So, it is highly desirable among the decision makers in the garment industry to track, analyse, and predict the productivity performance of the working teams in their factories.

Problem Statement

We want to build a machine learning model that can predict whether a garment industry will achieve its target productivity performance based on some parameters. We will measure productivity performance using a performance metric on a scale ranging from 0 to 1. The dataset used is garment-worker productivity, which is a public dataset because it is taken from the UCI repository website. We will build different machine learning models and compare the results between them.

Dataset

a. Data Collection

The dataset used in this study was published in 2020 with 15 attributes including date, day, quarter, department, team_no, no_of_workers, no_of_style_change, targeted_productivity, SMV, wip, over_time, incentive, idle_time, idle_men, actual_productivity with continuous actual_productivity classes has 1197 instances. In table below, we can see from the specification of the garments worker productivity dataset, which has 15 attributes and 1197 data.

No	Attribute	Description
1	Date	The date is in MM-DD-YYYY format
2	Day	Days of the week
3	Quarter	Part of this month. One month is divided into four parts
4	Department	The department is associated with the instance
5	team_no	The team number associated with the instance
6	no_of_wor kers	The number of workers on each team
7	no_of_style_change	The number of changes to a specific product style
8	targeted_ productivity	The targeted productivity is set by the Authority for each team for each day
9	SMV	Standard Minute Value, this is the time allocated for a task
10	wip	Work in Progress includes the number of unfinished items for the product
11	over_time	Represents the amount of overtime by each team in minutes
12	incentive	Represents the number of financial incentives (in the UDB) that enable or motivate certain actions
13	idle_time	The length of time the product has stalled for several reasons
14	idle_men	The number of unemployed workers due to production disruptions

15	actual_productivity	The actual percentage of productivity generated by workers. It ranges from 0-1

b. Exploratory Data Analysis

We performed some initial investigation on our dataset so that we can discover some patterns and insights about our dataset. This process helped us to spot anomalies, to test hypotheses and check assumptions with the help of summary statistics and graphical representations.

1. Is there any missing values and what are the data types of each feature?

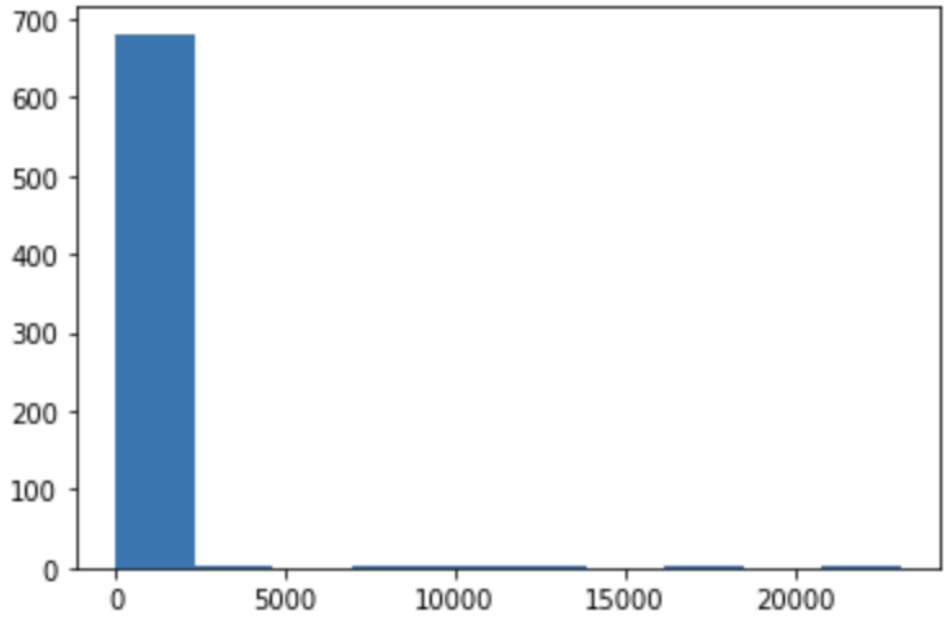
Process: We counted non-null value and type of each column

```
        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
```

Conclusion: Only ‘wip’ column has missing values, almost half of the values are missing. Also, the dataset has both numerical and categorical features.

2. As ‘wip’ has missing values, how to resolve this?

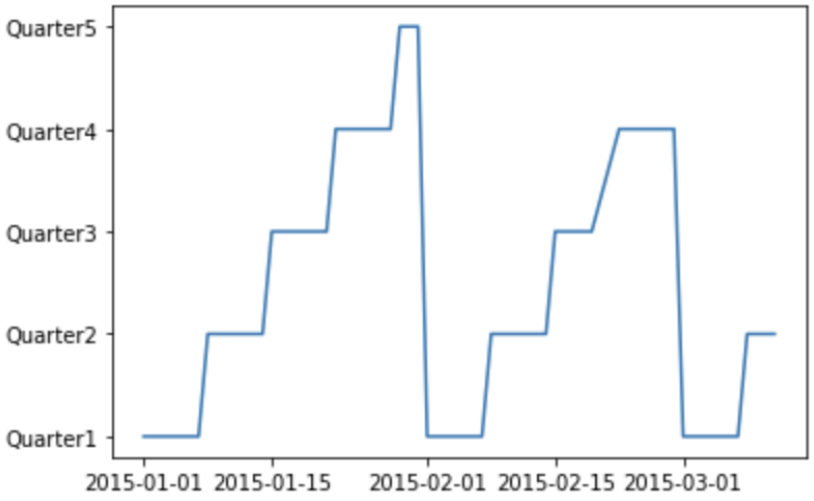
Process: Plotting the distribution of ‘wip’ will give us an idea about how to fill those missing fields.



Conclusion: Almost all of the values are 0, so we can fill the rows with the mode 0.

3. How to use the date object as a feature?

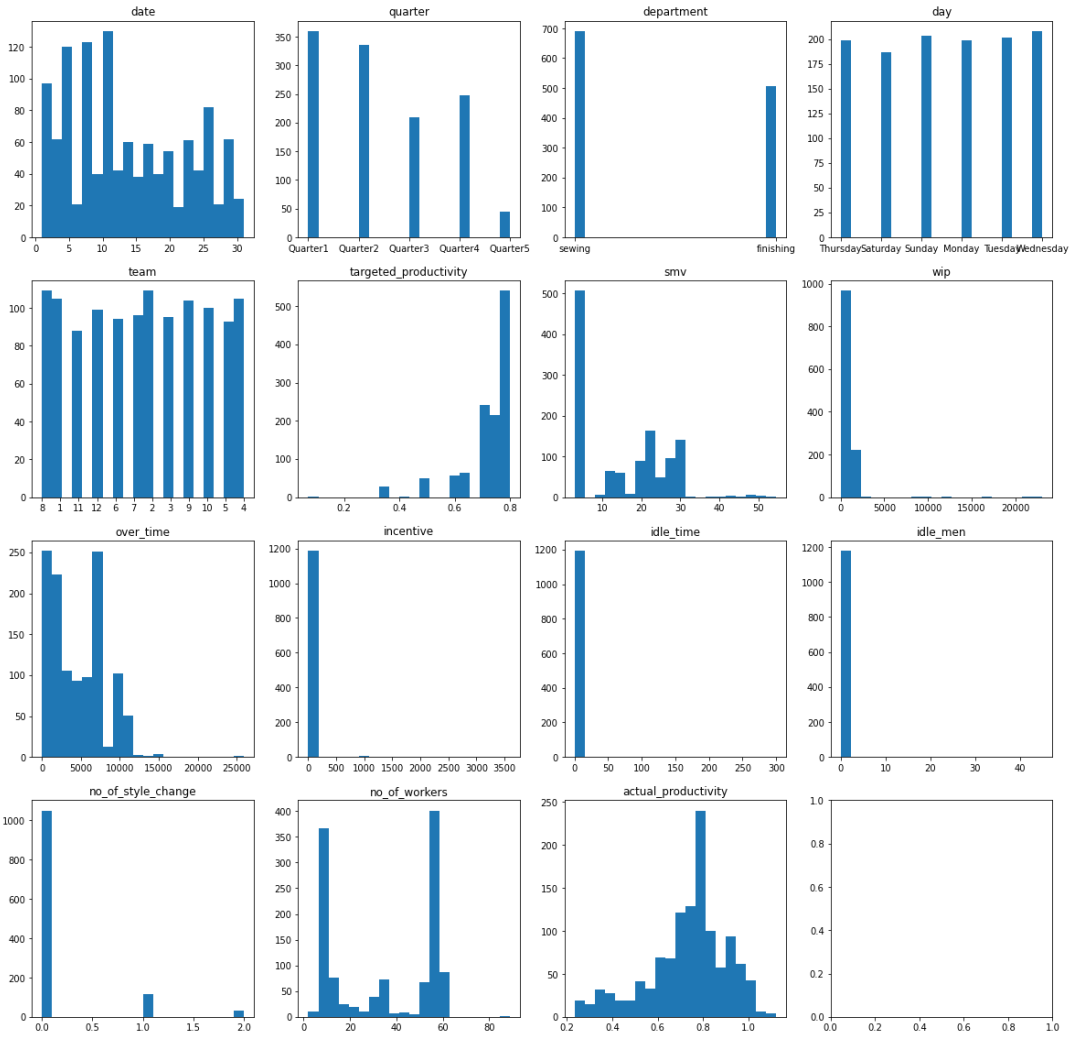
   Process: First we need to plot the ‘date’ column to understand the distribution.



Conclusion: Plotting date and found that all the data are from same year and there are only 3 different months, so we only extract the day of month as a numeric feature.

4. Is there any column that needs to be dropped?

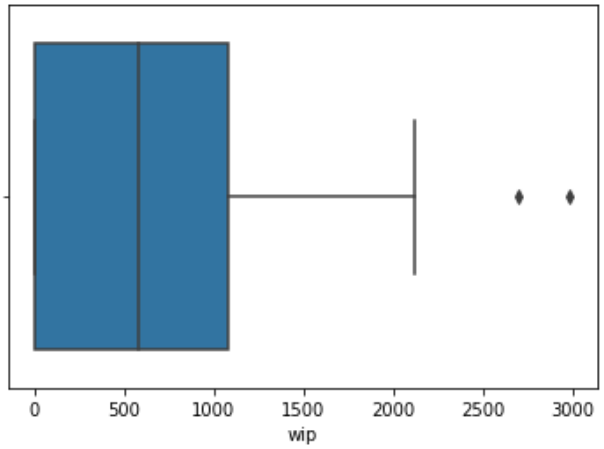
   Process: We plotted the data distributions for all the columns.



   Conclusion: From the distribution above *idle_men* and *idle_time* features seem to be useless for predicting actual_productivity, almost all of their values are zero and the rest few do not seem to produce any pattern with the other features. So we discard them. We also drop the date for now as it also shows no importance.

5. The ‘wip’ column may contain some outliers, how to resolve it?

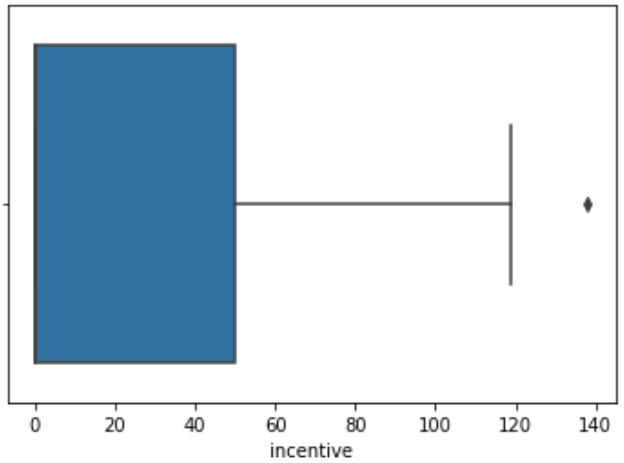
   Process: First we need to plot the boxplot of ‘wip’



Conclusion: The boxplot of wip indicates that they have huge outliers. We will drop those rows.

6. The ‘incentive’ column may contain outliers too.

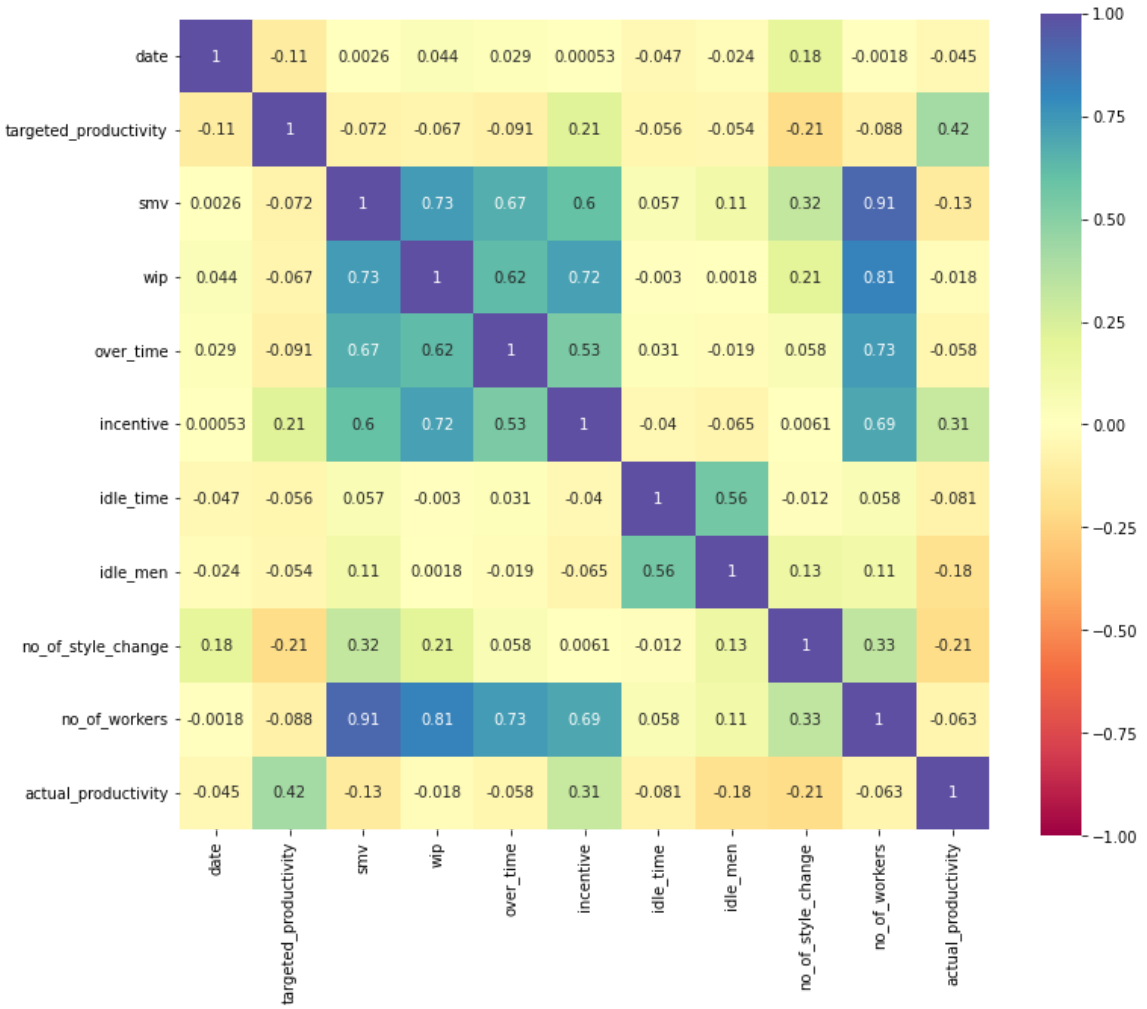
   Process: Likewise ‘wip’, we will observe the boxplot to understand if there are any outliers.



   Conclusion: It contains outlier. And the number of outliers is fairly small, so we can discard them.

7. Are all the features correlated to the target?

   Process: We generated correlation heatmap with colour-coded value, The least correlated variables could be considered as irrelevant and can be dropped.



Conclusion: The last row is the correlation between all other variables and the target (actual_productivity). As we can see date, wip, idle_time and no_of_workers are showing least contributions in predicting productivity. These columns can be dropped.

c. Data Preparation

We prepared our dataset in a few steps using the insights from the EDA part. To clean the data we followed these steps.

1. Null value fix: There was a significant amount of null values in ‘wip’ which we couldn’t possibly discard, but the overwhelming majority of ‘wip’ was 0 so we replaced all null values with 0.

2. Data type fix: ‘date’ was in object format, so we converted it to datetime. We also noticed that there were only 3 months’ data of the same year so we only kept the day of month.

3. Typo fix: In the ‘department’ column, ‘sweing’ was used instead of ‘sewing’, some rows contained an extra space at the end (‘finishing ‘ instead of ‘finishing’). We fixed them.

4. Outlier removal: The boxplots of ‘wip’ and ‘incentive’ showed that they have huge but very few outliers, so we discarded them.

5. Feature selection: Most values of idle_men, idle_time were zero. These columns along with date showed almost no correlation with our target variable, so we removed them.

6. One hot encoding: We used one hot encoding for the categorical features 'quarter', 'department', 'day', 'team'.

7. Normalization: We scaled the numeric features using the standard scaler which transformed mean to 0 and variance to 1.

Chapter 2

METHODOLOGY

Method Used

We tested three methods in our project. The rationale behind using these models are discussed in the following -

1. *Logistic regression:* We picked logistic regression as it is a type of regression analysis and is a commonly used algorithm for solving binary classification problems. Logistic regression is a classification algorithm that predicts a binary outcome based on a series of independent variables. In our project, this would mean predicting whether a garment industry could meet its expectations or not. Of course, logistic regression can also be used to solve regression problems, but it's mainly used for classification problems.

2. *Random forest*: Random forest algorithm can be used for both classifications and regression tasks and it provides higher accuracy through cross validation. It builds multiple decision trees and combines them together to get a more accurate result. This classifier will handle the missing values and maintain the accuracy of a large proportion of data. If there are more trees, it won't allow over-fitting trees in the model. Also it has the power to handle a large data set with higher dimensionality.

3. *Adaboost*: AdaBoost is best used to boost the performance of decision trees on binary classification problems, so we tested this model in this project. AdaBoost can be used to boost the performance of any machine learning algorithm. It is best used with weak learners. These are models that achieve accuracy just above random chance on a classification problem. The most suited and therefore most common algorithm used with AdaBoost are decision trees with one level. Because these trees are so short and only contain one decision for classification, they are often called decision stumps.

Random forest outperformed the others so we proceeded to do the rest of the training with it. Random forest is an ensemble learning method which aggregates the results of multiple decision trees but uses a random sample of features for splitting.

Training details:

1. Dataset split:
   We split the dataset into 60% training, 20% validation, and 20% test set.

2. Hyperparameters:
   Random forest classifier has 2 main hyperparameters - max depth & number of estimators. We used cross-validation grid search to jointly optimize these 2 hyperparameters. Out of the values we searched for, we got the best results using max depth = 8 and number of estimators = 128. number of estimators = 256 gave the same results, we chose 128 according to Ockham's razor principle.

Chapter 3

EXPERIMENTS

Results and Evaluation:

As our dataset was imbalanced, we had to use weighted f1-score as the evaluation metric. In the test set, our best model's results were:

	precision	recall	f1-score	support
False	0.84	0.84	0.84	96
True	0.89	0.89	0.89	140
accuracy			0.87	236
macro avg	0.87	0.87	0.87	236
weighted avg	0.87	0.87	0.87	236

Error Analysis:

For binary classification problems, there are two primary types of errors. Type 1 errors (false positives) and Type 2 errors (false negatives). It is often possible through model selection and tuning to increase one while decreasing the other, and often one must choose which error type is more acceptable. This can be a major tradeoff consideration depending on the situation.

In our case, the type 1 error is worse than the type 2 error. If a garment industry targets a particular productivity and the parameters indicate that they would achieve this target, for future benefits, they might make some bad decisions based on the prediction. And if the garment does not meet the goal eventually, it will face more trouble and loss.

Future Work:

Different pruning techniques can be tested. More data seems to be necessary to improve the model.

CONCLUSION

Even though it is possible to predict whether target productivity will be achievable or not in general cases with high accuracy, there are some hard-to-identify human factors (boredom, etc.) which can cause the productivity to fluctuate.

CODE

Github link: https://github.com/GitFazz/Fuse-Project-1