Assignment 3: Data Analytics

Using the *Productivity Prediction of Garment Employees Data Set*

Manuel Oberbacher*
Technische Universität Wien
Matr.Nr. 11831447
manuel.oberbacher@student.tuwien.ac.at

Thomas Laner†
Technische Universität Wien
Matr.Nr. 11807845
thomas.laner@student.tuwien.ac.at

For the purpose of this assignment, the data set utilised was the "Productivity Prediction of Garment Employees Data Set" which was obtained from the UCI Machine Learning Repository [3]. Utilising this data set allowed us to obtain a comprehensive understanding of the factors that affect productivity levels in the garment industry and to develop a model that can predict productivity levels with a high degree of accuracy.

1 BUSINESS UNDERSTANDING

1.1 Business Analytics Scenario

A garment manufacturing company is looking to improve the productivity of its working teams. The company has collected data on various factors that may impact productivity, including the number of employees on the team, the type of work being performed, the shift length, and any breaks taken. The company can use machine learning to predict the productivity based on these factors and identify patterns in the data that may indicate opportunities for improvement. This analysis could ultimately lead to increased efficiency and competitiveness in the market.

1.2 Business Objectives

The business objectives for this data set [3] are to increase productivity and efficiency in the garment manufacturing industry, and to meet global demand for garment products.

1.3 Business Success Criteria

For this work, we used four criteria to evaluate the proficiency of machine learning models: MAE, MSE, RMSE, MAPE. In-depth explanations for these metrics can be found in 4.1. Looking at the fact that all of these metrics indicate better performance with decreasing values - with 0 indicating a perfect performance, and because each model requires a different value range, we have chosen the values of these metrics based on the calculated baselines (see 5.3). Hence the lower bounds of requirements to our model are set to:

- MAE: 0.3
- MSE: 0.3
- RMSE: 0.55
- MAPE: 0.5

1.4 Data Mining Goals

The goal of data mining in the garment industry is to improve productivity by analyzing and predicting the performance of working teams in factories. This is done in order to increase competitiveness and profitability for the company by optimizing production and delivery performance.

1.5 Data Mining Success Criteria

In order to perform successful and helpful data mining, it is important to follow certain criteria. These include having clearly defined objectives, using high quality data, selecting appropriate algorithms, visualizing the data and results, considering ethical considerations and biases and continuously evaluating the process.

1.6 AI risk aspects that may require specific consideration

There are several AI risk aspects in the data mining process for the garment industry, including biases in the data and algorithms. These risks may lead to unfair or inaccurate assessments of productivity and may require specific measures to address them, such as ensuring representative and unbiased data and carefully reviewing and testing algorithms. Data privacy and security is also a potential risk that must be considered in the data mining process. It is important to address these risks in order to ensure the accuracy and fairness of the data mining process and to protect the privacy and security of workers and data.

2 DATA UNDERSTANDING: DATA DESCRIPTION REPORT PRESENTING

2.1 Attribute Types & Semantics

- **date:** A date attribute representing the date on which the data was collected, in the format MM-DD-YYYY. Type: *Ordinal.*
- **quarter:** A categorical attribute representing 7-day-intervals that are splitting each months with days 1-7 going into the first interval and so on. The possible values in the original data set [3] are *{Quarter1, Quarter2, Quarter3, Quarter4, Quarter5}*. Type: *Ordinal.*
- **department:** A categorical attribute representing the department associated with the data instance. The possible values are *{sweing, finishing}. Note: In the csv file there is a space included after the string finishing, so the type finishing appears 2 times. This inconsistency has been corrected in the pre-processing stage.* Type: *Nominal.*
- **day:** A categorical attribute representing the day of the week on which the data was collected. The possible values are *{Monday, Tuesday, Wednesday, Thursday, Saturday, Sunday}*. Type: *Ordinal.*

*Student A - Group 16
†Student B - Group 16

- **team:** A categorical attribute representing the team number associated with the data instance. The possible values are in the closed range *[1, 12]*. Type: *Nominal*.
- **targeted_productivity** : A numerical attribute representing the targeted productivity set by the authority for each team for each day. The possible values are in the range from 0 to 1. Type: *Ratio*.
- **smv:** A numerical attribute representing the standard minute value, or the allocated time for a task. Type: *Ordinal*.
- **wip:** A numerical attribute representing the work in progress, or the number of unfinished items for products. Type: *Ordinal*.
- **over_time:** A numerical attribute representing the amount of overtime by each team in minutes. Type: *Ordinal*.
- **incentive:** A numerical attribute representing the amount of financial incentive (in BDT) that enables or motivates a particular course of action. Type: *Ordinal*.
- **idle_time:** A numerical attribute representing the amount of time when the production was interrupted due to various reasons. Type: *Ordinal*.
- **idle_men:** A numerical attribute representing the number of workers who were idle due to production interruption. Type: *Ordinal*.
- **no_of_style_change:** A numerical attribute representing the number of changes in the style of a particular product. Type: *Ordinal*.
- **no_of_workers:** A numerical attribute representing the number of workers in the team. Type: *Ordinal*.
- **actual_productivity:** A numerical attribute representing the percentage of productivity that was delivered by the workers in comparison to the targeted productivity. The possible values are in the range from -0.61 to 0.1. Type: *Ratio*.

The table in Fig. 1 shows the data types as *pandas* recognizes them. As explained above, there are 15 types of data.
Note that the first four columns are recognized as general object types. This will be changed later on during the data preparation.

Fig. 2 shows a sample of 10 entries in the data set [3] that has been derived using the *pandas* command `df.sample(10)`.

2.2 Statistical Properties

The matrix in Fig. 3 shows the correlation between the numerical values in the raw data set [3]. In the following, correlations with at least a moderate degree are listed and possible reasons are explained:

- *smv* and *over_time* show a **strong correlation** of 0.67. This means that tasks that require a lot of time usually also require more overtime than tasks that require less time. This makes sense when you consider that longer tasks can cause more problems than short ones and therefore require more overtime.
- *smv* and *no_of_workers* show the highest correlation with a value of 0.91 which indicates a **very strong correlation**. This means that bigger teams usually work on tasks with a high amount of allocated time.

```
<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
```

Figure 1: Attribute types

index	date	quarter	department	day	team	targeted_productivity	smv	wip	over_time	incentive	idle_time	idle_men	no_of_style_change	no_of_workers	actual_productivity
1004	3/1/2015	Quarter1	finishing	Sunday	7	0.8	4.6	NaN	3360	0	0.0	0	0	8.0	0.350416667
1147	3/9/2015	Quarter2	sweing	Monday	7	0.5	30.48	1161.0	6600	0	0.0	0	1	55.0	0.500610909
914	2/24/2015	Quarter4	finishing	Tuesday	9	0.7	2.9	NaN	1800	0	0.0	0	0	15.0	0.726933333
459	1/27/2015	Quarter4	finishing	Tuesday	4	0.8	4.3	NaN	1440	0	0.0	0	0	12.0	0.940625
835	2/18/2015	Quarter3	sweing	Wednesday	9	0.7	18.79	1587.0	4560	30	0.0	0	1	51.0	0.70009573
424	1/25/2015	Quarter4	sweing	Sunday	1	0.7	22.94	1682.0	10530	75	0.0	0	0	58.5	0.800909609
185	1/11/2015	Quarter2	finishing	Sunday	11	0.8	4.15	NaN	1440	0	0.0	0	0	8.0	0.668087121
1105	3/7/2015	Quarter1	sweing	Saturday	5	0.7	27.48	541.0	6960	24	0.0	0	1	58.0	0.408960345
1126	3/8/2015	Quarter2	finishing	Sunday	10	0.7	2.9	NaN	960	0	0.0	0	0	8.0	0.441041667
724	2/11/2015	Quarter2	sweing	Wednesday	4	0.7	30.1	667.0	5940	40	0.0	0	1	59.5	0.700588406

Figure 2: Data set sample

- *over_time* and *no_of_workers* show a **strong correlation** with a value of 0.73. This means that larger teams spend more hours working over time than smaller teams, which could also be a result of the correlation that these two categories each have with *smv*.
  actual productivity and *target productivity* show a **moderate correlation** with a value of 0.42. This correlation suggests that productivity targets are more likely to be met if they are set at a higher value of the possible spectrum. We will analyse this result in more detail in the following sections.
- *idle_men* and *idle_time* show a **moderate correlation** with a value of 0.56. This correlation is due to the fact that some members of the workforce are unable to continue their work when there are problems in production. However, it must also be taken into account that some of the workers are then able to spend their time on another task - which is the cause of only a moderate correlation and not a high one.

2.3 Data quality

- **Completeness:** Through the use of *pandas*, it was possible to calculate the percentage of missing values in the data set. Thereby we could determine that 42% (506/1197) of the *wip* values of the attributes are NULL. Considering that *wip* actually represents the number of unfinished items of a given product, and looking at the remaining columns of a row where *wip* is NULL, it seems quite plausible that the missing entries actually represent finished products

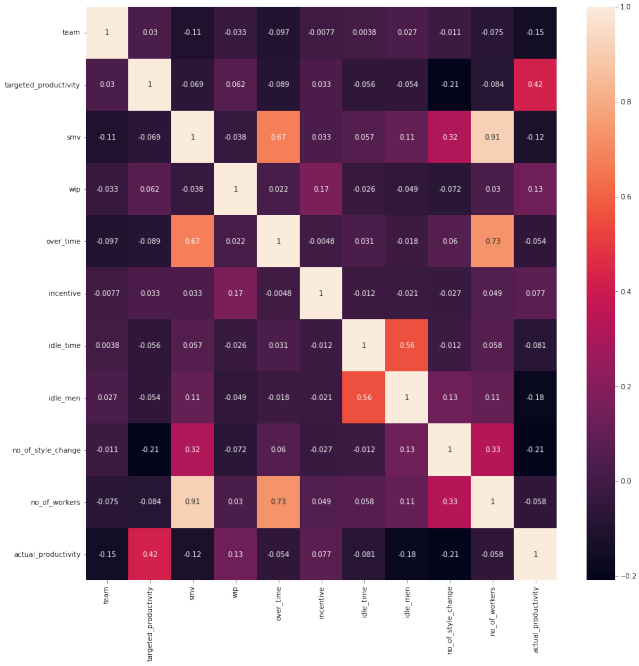

Figure 3: Correlation matrix

and should therefore be better represented with the integer 0 as the majority of the data set has *0* for this attribute. For the remaining attributes, all values are defined, so that completeness is inherently given for them.

- **Outliers:** Using the `describe()` function of *pandas* and the *Interquartile Range* (*IQR*), we found that several columns of the original data set [3] contained outlier data points (see Fig. 4). Due to our limited knowledge of how the data set [3] was collected, it is difficult to make assumptions about the nature of these outliers, but since our data analysis of the outliers did not reveal any noticeable traces and there seems to be no correlation between them, we decided to treat them as "noise" and therefore filtered out the ones in *over_time*, *wip* and *incentive* at a later stage.

```
Outliers in team = 0
Outliers in targeted_productivity = 79
Outliers in smv = 0
Outliers in wip = 22
Outliers in over_time = 1
Outliers in incentive = 11
Outliers in idle_time = 18
Outliers in idle_men = 18
Outliers in no_of_style_change = 147
Outliers in no_of_workers = 0
Outliers in actual_productivity = 54
```

Figure 4: Outliers per column

- **Inconsistencies of value representation:** As noted in 2.1, the attribute *department* has an inconsistency regarding a space after the category *finishing*. This could be clearly shown when printing all single values of the columns with non-numeric data types. Beyond that, we could not find any other data inconsistencies in the data set [3]. Since no explanation for the double presence of *finishing* could be found, both instances of this value are later combined into a single value.
  In addition, the *sewing* value in the *department* column was originally set to *sweing*. To prevent misinterpretation, we have changed these instances to the intended *sewing*.
  Another problem we discovered in the data set [3] is the naming of the column *quarter*. Since this column actually describes 7-day periods in each month, this choice of name is misleading. For this reason, we have renamed the column *seven_day_interval*. The values in the column do not need to be changed, as they are divided into numerical categories anyway.

2.4 Visual Exploration & Hypothesis

Looking at Fig. 5, it can be seen that the *sewing* department has significantly more staff members than the *finishing* department.

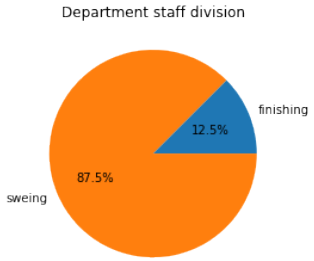

Figure 5: Division of staff members into the departments

Hence one would, also on the basis of a correlation between *smv* and *no_of_workers* think that the output of the *sewing* department is higher than the productivity of the *finishing* department. This thesis is supported if one looks at Fig. 6 which clearly shows that less working hours are spent in the *finishing* department by fewer workers compared to the *sewing* department.

Another thesis emerges from the examination of Fig. 3 and Fig. 7, which shows the actual and target productivity rate in percentage, is that as target productivity increases, so does actual productivity in a moderate correlation.

2.5 Ethical sensitivity evaluation

The given record does not contain any ethically sensitive information. This can easily be judged by the fact that there is no ethically sensitive information in the data types of the set.
Due to the fact that the department column has an imbalance compared to the data points of the *sewing* department (see **??**) has an imbalance, as out of 1141 entries only 484 have the value "*finishing*", while the remaining 657 entries in the column have the value "*sewing*", the *finishing* department can be considered a lower sample value compared to the *sewing* department.

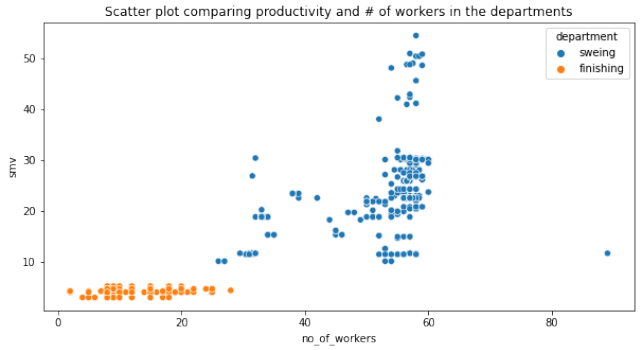

Figure 6: Comparison of the smv and number of workers in both departments

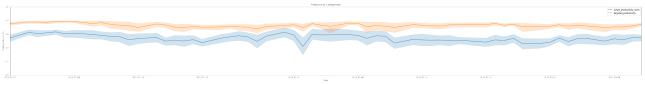

Figure 7: Actual and targeted productivity compared

2.6 Risk- & Bias analysis

Due to the structure of the data set [3], certain biases in the data analysis are possible. One of these could be in the form of confirmation bias, e.g. that larger teams work less efficiently. The risk of falling victim to these biases must therefore be kept in mind, especially when analysing graphical data representations, in order to avoid the danger of merely forming something from the data that does not exist in reality.

3 DATA PREPARATION REPORT

3.1 Exploring and Documenting options for Derived Attribute

Since the focus of this work is to produce a productivity analysis, we have added an additional column called *productivity_difference* which contains the difference between the targeted productivity (*targeted_productivity*) and the actual productivity (*actual_productivity*). To fill in this column, we used the following formula: $actual\_productivity - target\_productivity$. But this derived attribute has only been used for the scope of analysing the data set, since it is highly correlating with the actual productivity which we want to predict.

We also considered adding other columns, such as an incentive ratio, which contains the number of incentives required to perform a particular task, or a productivity per style change column, which indicates how much productivity is lost with each style change. However, these columns did not seem useful to us either for examining the dataset or for the machine learning that was added later, as the aim of this work was solely to predict actual productivity and not to look for reasons for possible deviations between target and actual productivity.

3.2 Exploring Additional External Data Sources

One possible data point that could be of interest in this analysis would be the average experience of workers in a team. Another data point that could be included is data describing external influences, such as data on the weather at the plant's location, economic data (e.g.: GDP, CPI, labour market data) and industry comparison data that shows how other companies in the region are doing and under what conditions the company has to work.

These data points could then be used to enhance the machine learning algorithms predictive power by giving it more information about the world which it tries to predict.

3.3 Pre-Processing

Our pre-processing consisted of several steps which had to be taken before applying a data mining algorithm to our data set [3].

3.3.1 Data Cleansing. is the first aspect that we want to consider in this regard. Therein, the first step that we took was to convert data points that were given in the *object* data type into fitting data types. Hence *date* and *day* (see 2.1) were converted into the data types *datatime64[ns]* and *int64* respectively. The final data types of the data set [3] can be seen below in Fig. 8.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1141 entries, 0 to 1196
Data columns (total 17 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   date                     1141 non-null   datetime64[ns]
 1   seven_day_interval       1141 non-null   category
 2   department               1141 non-null   category
 3   day                      1141 non-null   category
 4   team                     1141 non-null   category
 5   targeted_productivity    1141 non-null   Float64
 6   smv                      1141 non-null   Float64
 7   wip                      1141 non-null   Int64
 8   over_time                1141 non-null   Int64
 9   incentive                1141 non-null   Int64
 10  idle_time                1141 non-null   Float64
 11  idle_men                 1141 non-null   Int64
 12  no_of_style_change       1141 non-null   Int64
 13  no_of_workers            1141 non-null   Float64
 14  actual_productivity      1141 non-null   Float64
 15  productivity_difference  1141 non-null   Float64
 16  items_per_worker         1141 non-null   Float64
dtypes: Float64(7), Int64(5), category(4), datetime64[ns](1)
memory usage: 143.6 KB
```

Figure 8: Converted Attribute types

After converting the data types, we focused first on improving data quality and then on handling inconsistencies, categorising columns and lastly on encoding categorical data.

In order to improve the data quality of our set, we first handled the misleading column name, as it has been described in 2.3's *Inconsistencies of value representation*. Next, we handled missing values. According to our analysis (see 2.3's *Completeness*), the column *wip* exhibited 506 undefined values. As explained above, considering the structure of the data and the meaning of the data in the column *wip* (see 2.1), we have replaces the NULL values in the column with

the value 0.

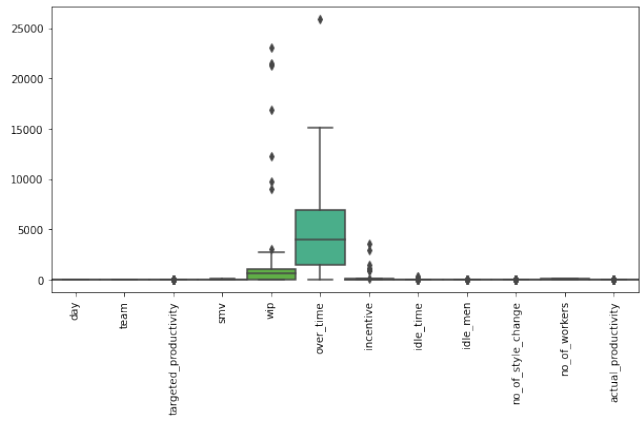

Figure 9: boxplot showing the outliers in the set

Following this step, we focused on the outliers that we were able to identify in the provided using boxplots (see Fig. 9) data set [3]. Hereby, the columns *over_time*, *wip*, *incentive* and *actual_productivity* exhibited outliers. We handled the outliers of the outliers in *over_time*, *wip* and *incentive* using the $1.5xIQR$-rule.
We dealt with the 35 outliers in the *actual_productivity* column that remained in the set after the handling of the outliers in the other columns by filtering out all values that were > 1 , as these would indicate an actual_productivity greater than the maximum possible productivity, which represents an impossible condition.
Below, a second boxplot, which has been created after the outlier handling can be seen in Fig. 10.
Next, inconsistencies in the data set [3] had to be handled. Namely,

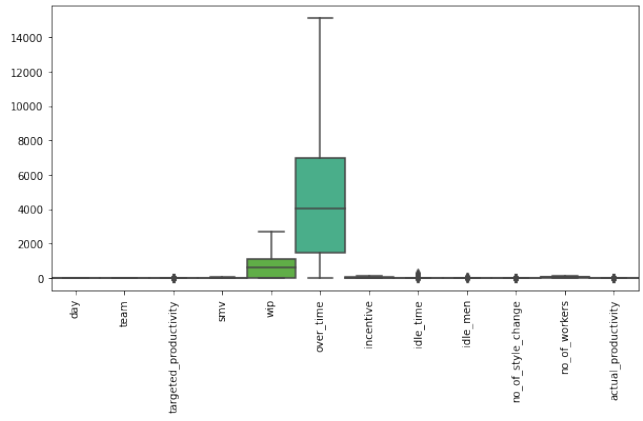

Figure 10: boxplot showing the set after outliers are handled

as described in 2.3's *Inconsistencies of value representation*, the value *finishing* in the column *department* had been represented in 231 of 484 instances with a trailing space. We handled this inconsistency by removing the trailing spaces in these instances.

3.3.2 Derived Attributes. are the next data preparation that we focused on in our work. Our exact work flow of this aspect, has been described in 3.1.

3.3.3 Transformations. are the next aspect that we have covered in our data preparation workflow. Thereby we categorised the columns *seven_day_interval*, *department*, *day* and *team* encoded them into numerical representations using the *OneHotEncoder* of the *scikit-learn* library.

3.3.4 Scaling. of the data in the set was performed in our workflow by using the *StandardScaler* of the *scikit-learn* library. As described in 4.1, we chose the random forest regressor for our model. While the Random Forest algorithm itself does not use the distance between points, standardizing the features can be helpful to improve the performance of the model.

3.3.5 Attribute removal. was the last preprocessing operation we performed to prepare the dataset for our machine learning model. We thereby removed the columns *date* and *productivity_difference*. . We removed *productivity_difference* because its value is not known in advance, as it depends on the value of *actual_productivity*. Therefore, the use of this derived attribute would provide information that is not known when used in a real environment.
The column *date* was removed as it could introduce a collinearity problem into our model, i.e. strong correlation with other attributes and difficulty in estimating coefficients in linear models or instability in tree-based models.

3.3.6 Binning. however, was not required in our case looking at the fact that the aim of our machine learning algorithm was not to categorise, but to predict values in form of a regression.

4 MODELING

4.1 Identification of suitable data mining algorithms

To find a suitable data mining algorithm for predicting actual productivity based on the given information (see 3), we evaluated different regression models using our training set (see 4.3). Using the metrics *MAE*, *MSE*, *RMSE* and *MAPE*, we were able to select the Random Forest Regressor as the most suitable algorithm for our needs.

- MAE (Mean Absolute Error) measures the average magnitude of the errors in a set of predictions without considering their direction. A lower value indicates a better fit of the model.
- MSE (Mean Squared Error) measures the average size of the errors, but also weights larger errors more heavily. A lower value indicates a better fit of the model.
- RMSE (Root Mean Squared Error) is the square root of MSE and is used to measure the average size of error in a range of predictions, with the same properties as MSE but in the same unit as the response variable. A lower value means a better fit of the model.
- MAPE (Mean Absolute Percentage Error) measures the average percentage error of the model and is useful when dealing with models that predict a percentage or proportion. A lower value indicates a better fit of the model.

The algorithms we tested are listed in the table below along with the resulting values of the metrics (see Fig. 11).

index	MAE	MSE	RMSE	MAPE
LinearRegression	0.09213	0.01657	0.12874	0.15867
KNeighborsRegressor	0.10130	0.02017	0.14202	0.17539
SVR	0.09956	0.01923	0.13868	0.17955
DecisionTreeRegressor	0.07439	0.01575	0.12551	0.12315
RandomForestRegressor	0.07044	0.01287	0.11343	0.12518
GradientBoostingRegressor	0.07083	0.01214	0.11020	0.12772
AdaBoostRegressor	0.08807	0.01549	0.12444	0.15011
Ridge	0.09234	0.01658	0.12878	0.16014
BayesianRidge	0.09296	0.01681	0.12967	0.16222
MLPRegressor	0.11496	0.02756	0.16601	0.19361

Figure 11: Regressors assessed on our test set using different metrics

Based on this evaluation, we selected the random forest regressor as a model and worked with it from then on.

4.2 Identification of suitable hyper-parameters

The random forest algorithm that we have chosen for our model has several parameters that can be adjusted. We chose the following parameters to tune our model:

(1) *n_estimators*: The number of trees in the forest.
(2) *max_depth*: The maximum depth of the tree.
(3) *min_samples_split*: The minimum number of samples required to split an internal node.
(4) *min_samples_leaf*: The minimum number of samples required to be at a leaf node.
(5) *max_features*: The number of features to consider when looking for the best split.
(6) *bootstrap*: If True, individual trees are fit on a random subset of the training data.

However, it must be mentioned that there would have been many more parameters that could be adjusted that influence the performance of the algorithm.

4.3 Train- / Validation- / Test set

We split our pre-prepared data set [3] using *scikit-learn*'s *train_test_split* function into:

- 60% train data
- 20% validation data
- 20% test data

4.4 Train-set-training and hyper-parameter tuning of model

In order to perform the initial training of our model, we have used the train-set which was specified in 4.3. In the following, we then tuned the hyper-parameters of our model which we identified in 4.2. Thereby we used the following parameters:

- *bootstrap*: [True]
- *max_depth*: [2, 4, 10, None]
- *max_features*: [4, 12, 15]
- *min_samples_leaf*: [1, 4]
- *min_samples_split*: [2, 5, 10]
- *n_estimators*: [1, 10, 100, 400]

4.5 Performance metrics analysis

The following two graphs (Fig. 12 and Fig. 13) show the tuning process of the selected hyper parameters (see 4.2).
Fig. 12 shows the test score of the model. Therby a higher mean test score indicates that the model is performing better on the test set, and therefore is more likely to perform well on unseen data.

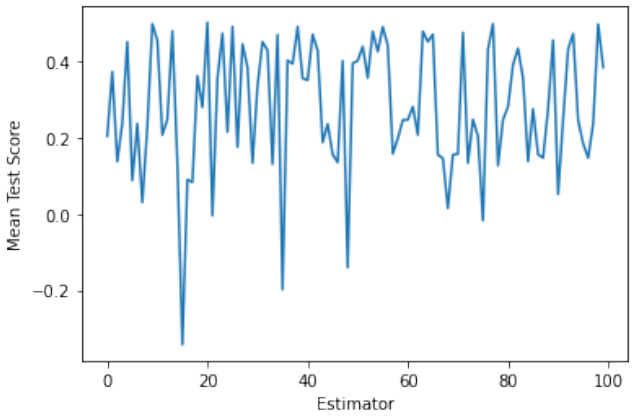

Figure 12: Mean test score

Fig. 13 depicts the standard deviation of the test scores. In this representation, a lower std test score indicates that the model's performance is more consistent across different folds of cross-validation, which means that the model is less likely to overfit to the training data.

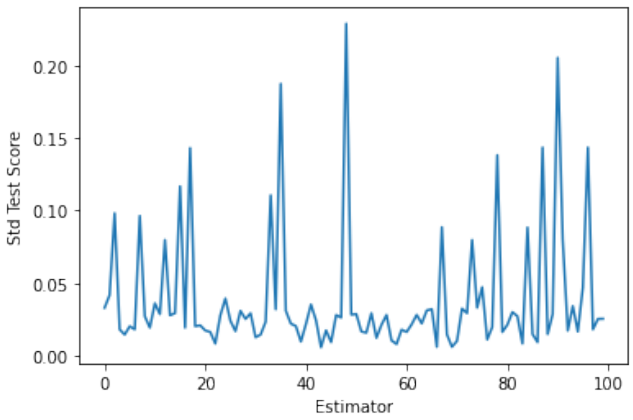

Figure 13: Standard deviation of test score

Based on the analysis that was presented above, we have used the following parameters:

- *bootstrap*: True
- *max_depth*: None
- *max_features*: 15
- *min_samples_leaf*: 1

- *min_samples_split*: 2
- *n_estimators*: 400

5 EVALUATION

5.1 Performance analysis of final models' application on test data

After applying the model using the hyper-parameters defined in 4.5 to the test data, the following performance was measured:

- MAE: 0.08133055393796487
- MSE: 0.01765226700540315
- RMSE: 0.1328618342693008
- MAPE: 0.14246647899336667

An explanation of the used performance metrics can be found in 4.1.

5.2 Performance analysis of model re-training

After re-training the model with the hyper-parameters defined in 4.5 with the full training and validation data, the following performance was measured we noticed an improvement in the prediction of the test data:

- MAE: 0.07693815473891227
- MSE: 0.016330855171862224
- RMSE: 0.12779223439576531
- MAPE: 0.13411305864312104

An explanation of the used performance metrics can be found in 4.1.

5.3 Literature review & baseline performance

In our literature search, we identified two relevant studies, Balla et al. [1] and Imran et al. [2], which used the same data set [3]. Balla et al. [1] evaluated the performance of different data mining algorithms and obtained the results shown in Figure 14. Imran et al.

Algorithm	*Correlation Coeficients*	MAE	RMSE
Random Forest	0.7071	0.0787	0.1236
Regresi Linier	0.5173	0.1081	0.1494
Neural Network	0.4169	0.1218	0.1763

Figure 14: algorithm test results of [balla]

[2], on the other hand, used a deep neural network and achieved the performance shown in Figure 15. Based on the results of these

	MSE	**MAE**	**MAPE**
Training	0.020	0.091	16.883
Validation	0.018	0.089	15.949
Test	0.018	0.086	15.932

Figure 15: performance of model [imran]

studies, we can set the performance of a random forest algorithm as reported by Balla et al. [1] for the data set [3] as a benchmark for our own study.

To determine the baseline performance of a regression model for the data set [3], we calculated the mean of the outcome variables and used it to calculate the following evaluation metrics: MAE, MSE, RMSE and MAPE (as described in section 4.1). The resulting values are as follows:

- MAE: 0.13043953952354573
- MSE: 0.028139250226980245
- RMSE: 0.1677475789005023
- MAPE: 0.2300838010276633

These values serve as a benchmark for comparison with the performance of any regression model applied to the data set [3].

5.4 Benchmark and baseline performance comparison

After evaluating the performance of our model using the metrics presented in section 4.1, a comparison was made with the benchmark and baseline performances.
The results show that the performance of our model is within an acceptable range. In particular, the values of the Mean Absolute Error (*MAE*), Mean Square Error (*MSE*), Root Mean Square Error (*RMSE*), and Mean Absolute Percentage Error (*MAPE*) metrics suggest that our model has a better fit compared to the baseline and a slightly better fit compared to the benchmark.

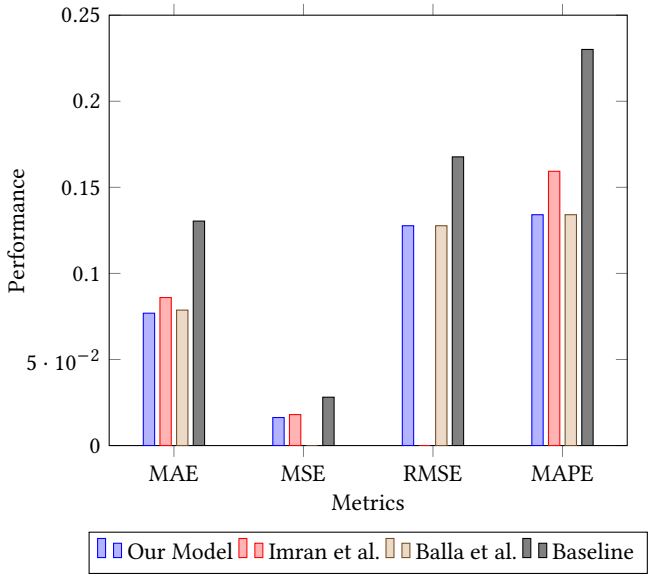

Figure 16: Performance comparison of baseline, benchmarks and our model

In Fig. 16, these different metrics are presented for our model (blue), [2]'s model (red), [1]'s model (brown) and the baseline which was calculated using the mean (grey).

5.5 Validation of performance using business success criteria

The metrics of the model (see 5.2) are all within the ranges specified in the business success criteria (see 1.3). Therefore, the performance of the random forest regression model (see 4) can be considered acceptable according to the business requirements and can be deployed in an appropriate manner under rigorous testing and performance evaluation requirements (see 6).

5.6 Protected attribute & bias analysis

Considering that the department attribute has an imbalance against the *sewing* data points (see 2.5), we analysed our model in more detail depending on these two values.
To do this, we split the data set [3] into two data frames based on their value in this column. This analysis allowed us to extract the following metrics:

Sewing:

- MAE: 0.032378261425219546
- MSE: 0.004120160974567458
- RMSE: 0.06418848007678214
- MAPE: 0.0664257328786742

Finishing:

- MAE: 0.1291368869063809
- MSE: 0.030634811231550388
- RMSE: 0.17502802984536617
- MAPE: 0.2134039259671873

Looking at these values and the results of the metrics presented in 5.2, we can assume that the model is strongly influenced by the data of the sewing department.
If only the performance of the *finishing* department were considered, our model would be only slightly better than the baseline calculated using the mean of the whole data set. However, if we were to calculate the baselines of the re-split data sets, we would get the following baselines:

Sewing Baseline:

- MAE: 0.10673894508824115
- MSE: 0.021054062621274114
- RMSE: 0.14510018132750252
- MAPE: 0.19316329559367462

Finishing Baseline:

- MAE: 0.1555608603776871
- MSE: 0.03580918138993155
- RMSE: 0.18923314030563343
- MAPE: 0.27165568137704427

Using these new baseline values, it is possible to see that our model still performs better than the baseline in all cases.

In summary, our model is better trained to predict *actual_productivity* in the *sewing* department and therefore the predictive power of our model is also better when used in the *sewing* department compared to the *finishing* department.

6 DEPLOYMENT

6.1 Comparison of performance with business objectives

Based on the results of our model (see 5), we would recommend that the model be used in a hybrid approach. This would combine both automated and manual processes and decision-making. This approach can provide the benefits of automation, such as increased efficiency and reduced error rates, while allowing for human control and decision-making. For example, the model could be used to automatically generate predictions of actual productivity, but a team of experts could review and verify these predictions before they are used to make decisions and take action.
It is also important to remember that the performance of the model may vary depending on the specific data space in which it is applied. For example, the model may perform well on one subset of data but less well on another (see 5.6). Therefore, it would be beneficial to evaluate the performance of the model for different subsets of data and possibly only apply it in the areas where it performs well (e.g.: only deploy it in the *sewing* department and not in the *finishing* department).

6.2 Ethical aspects & risks in deployment

Several ethical considerations must be taken into account when applying the developed model.
One potential problem is the possible bias of the model. The model's predictions may be based on historical data that is not representative of the current workforce or favours certain groups. One of these biases has already been identified in 5.6.
Another ethical issue is the possibility that the model displaces human workers. For example, automating productivity prediction may eliminate jobs previously held by human workers. Therefore, it is essential to consider the impact of job displacement on the workforce and to provide support and retraining opportunities for affected workers.
In the impact assessment, it is crucial to consider the potential risks and benefits of using the model. For example, the model may improve productivity and efficiency but also lead to job losses. Therefore, it is essential to weigh these potential risks and benefits and consider any unintended consequences of adoption.
As mentioned earlier (see 6.1), a hybrid solution for deployment could be to use the model to support human workers in making forecasts. This approach can help mitigate the risk of job displacement while reaping the benefits of the model.
In addition, it is advisable to continuously monitor and evaluate the performance of the model and its impact on the workforce. This can be done by gathering feedback from workers and managers and analysing data on productivity and job displacement. Further analysis may also be necessary to ensure no further biases in the model.

6.3 To-be-monitored aspects and intervention triggers

To ensure that the model performs as expected during deployment, it is important to monitor several key aspects of the model's performance. One important aspect to monitor is the overall performance of the model in terms of its ability to accurately predict actual productivity. This can be measured using the metrics mentioned earlier by comparing them to the business objectives and success criteria for the model.

Another important aspect to monitor is the performance of the model in specific subsets of the data, e.g. in different departments within the apparel industry. As mentioned in 5.6, we have already identified such an imbalance. If it becomes apparent that the model is performing particularly poorly in these or other subsets of the data, it may be necessary to intervene (e.g.: obtain more data on other departments) and make adjustments to the model to improve its performance in these subsets.

Triggers for intervention during deployment could be a significant deviation from expected performance, a change in the distribution of the data, or a particularly high error rate in certain subsets of the data (with a particular focus on the finishing department). When any of these triggers occur, it may be necessary to intervene and make adjustments to the model to improve its performance and better meet business objectives.

In addition to monitor performance and triggers during deployment, it is also important to further analyse the model and its performance after deployment to identify areas for improvement and make updates as necessary. This could include updating the model parameters, re-training the model with new data or adding new features to the model.

6.4 Reproducibility assessment

This assignment emphasised the concept of reproducibility throughout the entire research process. This is particularly crucial when dealing with machine learning techniques, as these models depend highly on the underlying data. To ensure the reproducibility of the results, the random state of the algorithm was set and maintained throughout the analysis. Furthermore, the methodology and code used in the analysis were documented in the form of a Jupyter notebook, which was made available along with this report.

The Jupyter notebook includes all the necessary information and commands used to obtain the data, including the source of the data set and the steps for statistical analysis, data cleaning and preprocessing. Additionally, the data set was split into train, validation and test sets. This process was vital to guarantee the robustness of the results and the generalisability of the findings to other similar data sets.

7 SUMMARY OF FINDINGS

7.1 Overall findings & lessons learned

The present assignment aimed to assess the ability of the *RandomForestRegressor* to predict actual productivity in the garment industry, despite the low correlation of the actual productivity values with other attributes. Through the analysis presented in the subsection 5.6 it was shown that the prediction accuracy of the model was significantly better for the *sewing* department, compared to the *finishing* department.

Despite the relatively low correlation between actual productivity and other attributes, the results indicate that the *RandomForestRegressor* is capable of predicting actual productivity with a high degree of accuracy. However, it should be noted that a more robust evaluation of the model's performance would have required the use of a stratified sampling technique to split the training set. Unfortunately, due to the limited size of the data set, this approach was not possible.

This finding highlights the importance of having a sufficient amount of data for training machine learning models, as well as the need for careful consideration of the sampling technique when splitting the data set. Overall, the results of this assignment suggest that the *RandomForestRegressor* has a good level of accuracy for the majority of the data, but there is still room for improvement.

REFERENCES

[1] Imanuel Balla, Sri Rahayu, and Jajang Jaya Purnama. 2021. Garment employee productivity prediction using random forest. *Jurnal Techno Nusa Mandiri*.

[2] Abdullah Al Imran, Md Nur Amin, Md Rifatul Islam Rifat, and Shamprikta Mehreen. 2019. Deep neural network approach for predicting the productivity of garment employees. In *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, 1402–1407. DOI: 10.1109/CoDIT.2019.8820486.

[3] 2020. Productivity Prediction of Garment Employees. https://archive.ics.uci.edu/ml/datasets/Productivity+Prediction+of+Garment+Employees. Retrieved Jan. 28, 2023 from.