Workers Productivity Prediction

Shreyank Buddhadev

October 2021

1 Problem

Given dataset of garment workers productivity predict actual productivity of workers.

1.1 Dataset Overview

Below table depicts feature names and type of dataset features.

feature	type
quarter	categorical
department	categorical
day	categorical
team	categorical
targeted_productivity	numerical
smv	numerical
wip	numerical
over_time	numerical
incentive	numerical
idle_time	numerical
idle_men	numerical
no_of_style_change	numerical
no_of_workers	numerical
actual_productivity	numerical

Provided dataset contains **14** features and **1160** data points.

1.2 Data Preprocessing

Provided data had several inconsistencies to overcome those below mentioned steps are applied on data set. To handle inconsistencies there were several possible alternatives, this section contains approach which were best suited to achieve least error.

1.2.1 Filling NaN values

Below depicted features has provided NaN values.

feature	Number of NaN
targeted_productivity	384
wip	703
over_time	371
quarter	353
day	375

- *wip(work in progress)* column has 61% NaN values. But further analyzing dataset below were the findings:
  - *wip* values for rows having *department* ***finishing*** are NaN and that accounts for 71% of total missing values
  - Assuming that, *department* ***finishing*** might have some waiting time before work in progress. All NaN's with *department* ***finishing*** are replaced with zeros.
  - for remaining 21% of missing values has *department* ***sweing***, statistical analysis for which is given below,

    mean 1150.280088
    min 7.000000
    median 1035.000000
    max 23122.000000

    There are some outliers as depicted in the figure 1. Also, standard deviation is high therefore mean might be deviated from central value because of outliers.So, remaining NaN's are replaced with median value of *wip* with *department* ***sweing***.
- *targeted_productivity* feature has numerical data. To handle NaN values in it we've statistically analyzed it and found below results.

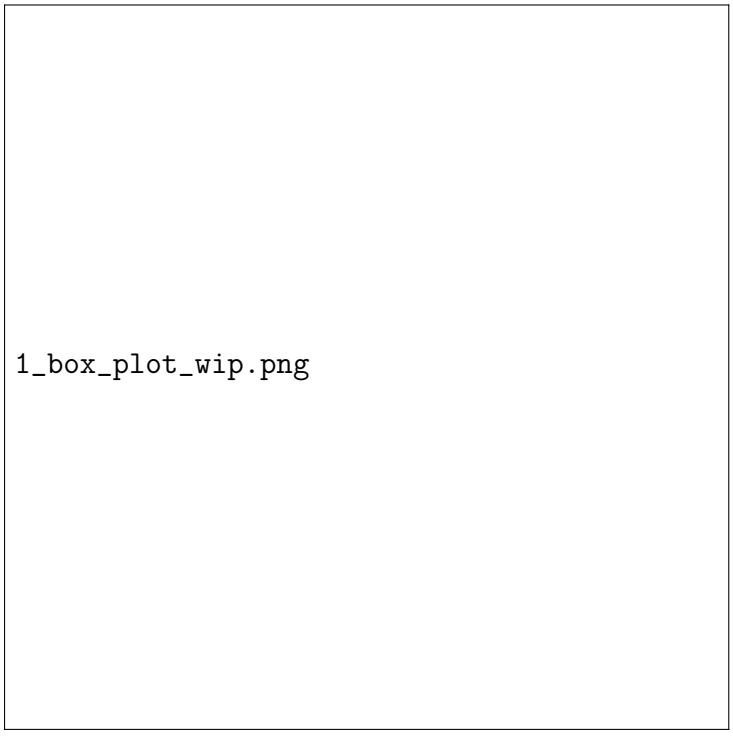


Figure 1: wip box plot

- mean: 0.732668
- median: 0.750000
- min: 0.350000
- max: 0.800000
- Since, mean and median has not much difference between them for *targeted_productivity* column. All Na N's for feature *targeted _productivity* are replaced with mean value of available *targeted _productivity* values.

- Feature *overtime* has numerical data. Below is statistical analysis for the feature
  - count 789.000000
  - mean 4612.053232
  - std 3390.363700

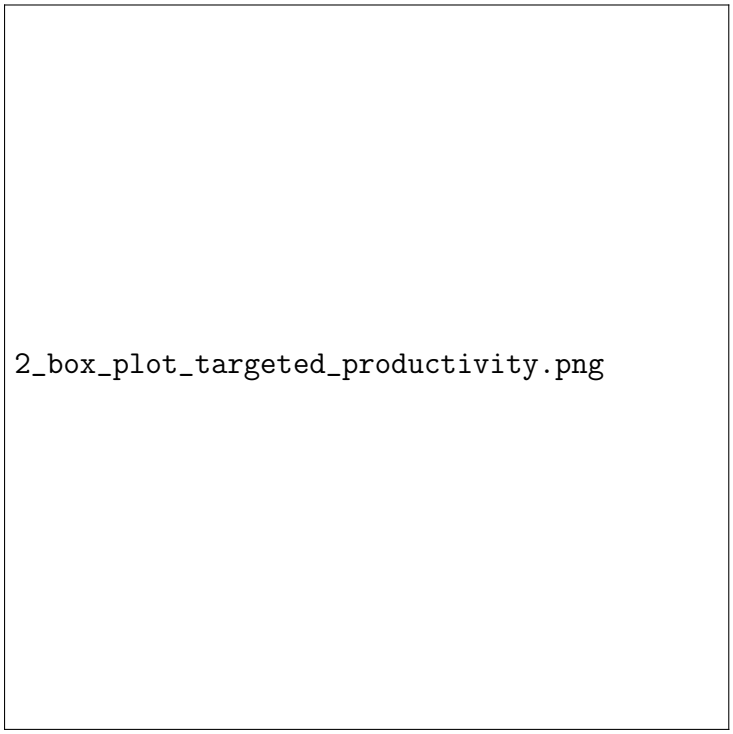


Figure 2: targeted productivity box plot

  - min 0.000000
  - median 4080.000000
  - max 25920.000000
  - *overtime* feature's data has some large values which contributes to higher deviation. But, common sensibly we can think that all the NaN's corresponds to 0 which translates that each department workers might not be required to do overtime.

- Feature *quarter* and Feature *day* has categorical data. To fill NaN's several possible approaches can be
  - to fill with most frequent value
  - to simply ignore NaN's
  - to assign a new categorical value

– Analyzing above possibilities, NaN Values for Feature *quarter* and *day* were replaced by new categorical values.

1.2.2 Encoding features

To fit categorical data into numerical model encoding has been applied.

- Feature *quarter, department* and *day* has categorical data which were encoded.
- Feature *quarter, day* had multiple unique values. So, to avoid adverse impact of it on model it was encoded using one hot encoding.
- Feature *department* had only 2 values. So, it was encoded using label encoder.

1.2.3 Normalizing or Standardizing data

- Normalization process $x_i = (x_i - min)/(max - min)$
- Standardization process $x_i = (x_i - \mu)/\sigma$

  here x_i = data point, μ = mean, σ = standard deviation
- Among above two approach, least square error difference was negligible.

1.2.4 Linearity and Multi colinearity

Below heatmap shows linear independence between columns. From the heatmap we can observe that *no_of_workers* has high co relation with *smv*.

1.3 Linear Regression loss function plots

figure 4b, 4a, 5a and 5b contains graph for gradient descent mean sqaure error, gradient descent mean absolute error, newton method mean absolute error and newton method mean square error respectively.

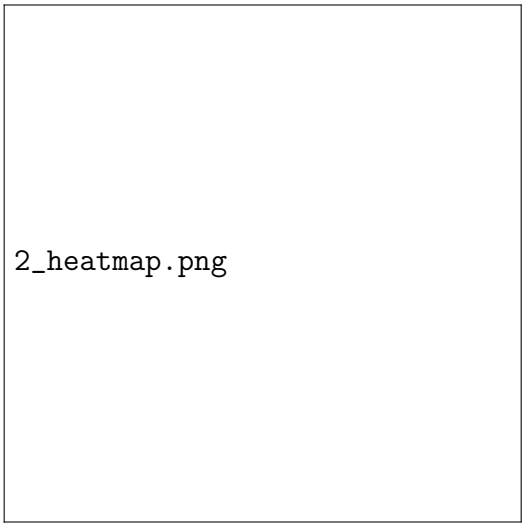


Figure 3: heatmap for features of given dataset

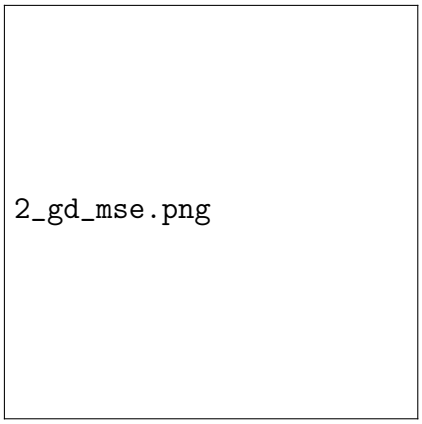
1.4 Accuracy Analysis

For linear regression

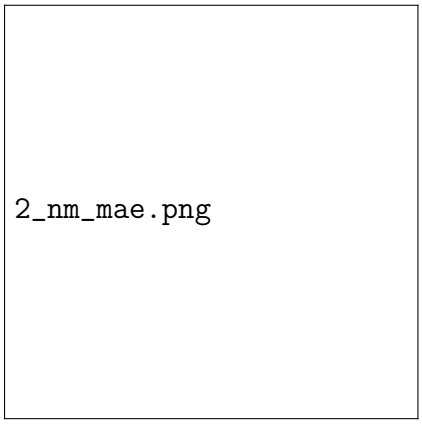
type	mean square error	mean absolute error
gradient descent-training	0.01283256	0.11742528590710881
gradient descent-testing	0.01108155	0.1157965155900566
newton's method-training	0.01201337	0.11387367756711504
newton's method-testing	0.01004012	0.11076702506485948

For newton method 1000 iteration were taken while for gradient descent 10000 were taken.

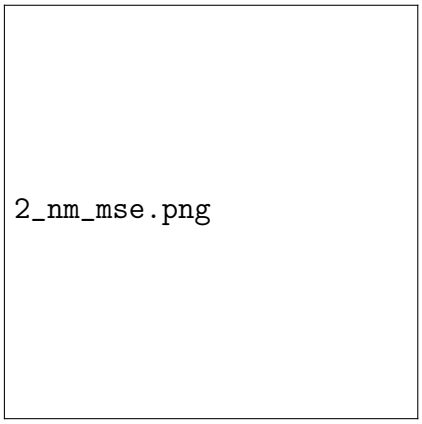
(a) gradient descent mae



(b) gradient descent mse



(a) newton method mae



(b) newton method mse