Predicting the productivity of garment factories

3804ICT Group 7

Group List:
Sebastian Perry (S5132483)
Josh Pearson (S5177636)
Bennett Taylor (S5095512)

1. At Glance

Data mining is one of the fastest growing fields in information technology, with good reason too. As technology advances, the amount of data collected increases. This makes data mining an integral part of the evolution of information technology. The aim of data mining is to uncover patterns or valuable knowledge from a given collection of data. For example, Google’s *Flu Trends* tries to locate hotspots for flu activity based on user’s search terms. This shows how data mining can assist current global challenges by extracting knowledge from large collections of data. It can also help with business intelligence by acquiring a better understanding of the commercial context, such as their competitors, supply/resources, the market and their customers. Machine Learning is a field that greatly benefits data mining. Machine Learning is the study of creating computer programs to recognize complex patterns and make intelligent decisions based on data. Supervised Learning (a branch of machine learning) will be used to classify and predict the productivity of employees.

The garment industry is one of the most dominating industries in today’s globalized society. It plays a key role in the growth of a country's economy by generating employment and trade. Its large global demand therefore requires maximum productivity for each company in the field. Since the industry heavily depends on human labour for production, its productivity is directly correlated to its employees. This means that things such as employee count, amount of work in progress, has a large effect on productivity. This leads to a problem in the industry where the actual productivity of an employee does not meet the targeted productivity set by the business to meet production goal deadlines. The business can face huge losses when this productivity gap occurs. This project aims to shorten that productivity gap by predicting actual productivity of employees.

2. Data Description

In order to make these predictions a dataset is needed which could be used to find the optimal variables which could maximise productivity. It was decided that information on the subject would be sourced from the UCI Machine Learning Repository as it had a perfect dataset consisting of 15 attributes and 1197 instances. Although it does have some issues with missing and messy data, it should be relatively easy to remove these issues while still preserving the insights such a dataset can supply. The dataset in question records the productivity of various garment employee teams, along with an array of other relevant attributes (at the time of recording) which will hopefully allow for the creation of a link between the most efficient variable combination required in order to achieve maximum productivity.

(please note: for a more indepth break down on the specifics of the attributes, please refer to appendix 1)

3. Algorithms and Techniques

3.1. K Nearest Neighbours

K Nearest Neighbours (otherwise known as k-NN) is a classification and regression algorithm commonly used in machine learning and statistics. It is known for its simplicity and its effectiveness at classification. Unlike other classification models, no training is needed; for each case, the algorithm compares the datapoint to the training set.

The concept of the algorithm is that it compares the distance between the data points to all of the training set. The algorithm then chooses k (k can be any number) number of nearest data points in the test set and assigns the most frequent class to the original data point.

The distance function is crucial, and having a distance function that performs well with a specific data set will lead to more accurate classifications. Examples of distance functions include Euclidean distance and Manhattan distance, but in this project we will compare different distance functions to find the most optimal one for the garment dataset.

3.2. Bayes Classification

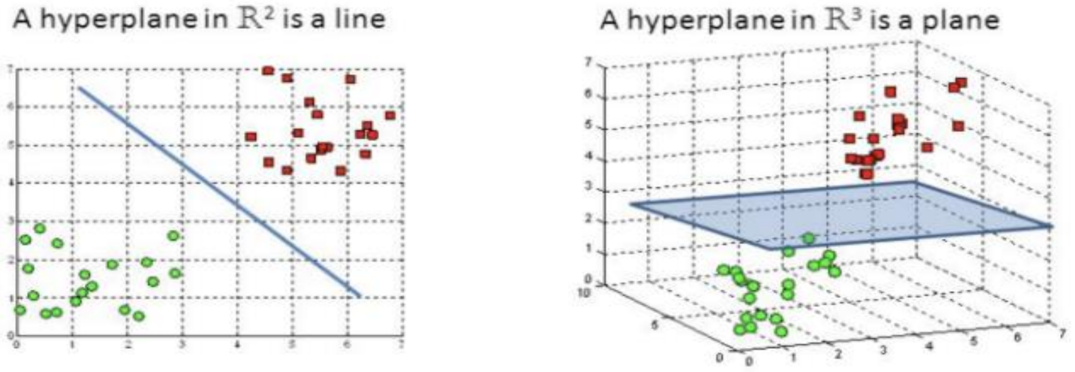
Bayes classification is a probabilistic prediction algorithm. It is based on Bayes theorem, and is known to have comparable performance with decision trees and neural networks. The formula for Bayes theorem is:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Bayes theorem is based on posterior probability, meaning it calculates the probability that the dimension results in a certain classification based on prior known knowledge. The prior known knowledge in this case is the probabilities calculated from the training set.

3.3. Support Vector Machine

Support Vector Machines (SVM's) are a machine learning algorithm used for classification and regression of both non-linear and linear data. It is highly preferred in many situations because it produces significant accuracy with low computational power. The aim of SVM's is to transform the original training data into a N-dimensional space (with N being the number of features) in order to search for a linear optimal separating hyperplane that distinctly classifies the data.



Hyperplanes in 2D and 3D feature space

There are many hyperplanes that could be chosen, but the optimal one is found at the maximum distance between the data points of both classes. The SVM finds this hyperplane by splitting the data into support vectors. These support vectors are data points that influence the position and orientation of the hyperplane due to how close they are to the hyperplane.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases} \qquad c(x, y, f(x)) = (1 - y * f(x))_+$$

Hinge loss function (function on left can be represented as a function on the right)

SVM’s use the hinge loss function to help maximize the distance between data points and the hyperplane. If the predicted value and the actual value are of the same sign then the cost will equal 0. If they are not, then a regularization parameter is added to the cost function to balance the distance maximization and loss.

3.4. Decision Tree

A decision tree is a classification algorithm. The algorithm is easy to interpret and inexpensive to construct.

A training set of data is used to construct a decision tree, and each branch holds a question. At the leaves, the data is then classified. In order for a question to be constructed, there must be certain information gained from asking the question. This reduces the chance of overfitting the model to the training set.

4. Evaluation Measures (10 points)

From measuring efficiency to the compactness of a model, there are many ways to compare classification models. Due to our project using multiple classification methods, there is a need to use a standardised metric to compare. The methods of evaluation between the different models will be accuracy, precision, recall, and F1-score.

	Classified Positive	**Classified Negative**
Actual Positive	True Positive(TP)	False Negative(FN)
Actual Negative	False Positive (FP)	True Negative(TN)

Accuracy

Accuracy (a) is calculated by dividing the number of correctly classified examples by the total number of samples.

$$a = \frac{TP + TN}{TP + TN + FP + FN}$$

Whilst the measure is not favourable to highly skewed or imbalance data, it is a commonly understood and relatable metric, therefore we will consider it when evaluating the algorithms

Precision

Precision(p) is calculated by dividing the number of correctly classified positive examples by the total number of examples that are classified as positive.

$$p = \frac{TP}{TP + FP}$$

It is fraction of relevant instances among all retrieved instances

Recall

Recall (r) is calculated by dividing the number of correctly classified positive examples by the total number of actual positive examples in the set.

$$r = \frac{TP}{TP + FN}$$

It is the fraction of retrieved instances among all relevant instances

F1-value

The F1 value is a combination of precision and recall, and combines it into one measure. Whilst the measures above are good for comparison, this final F1 value will be the main measure of evaluation amongst the different classification algorithms.

$$F_1 = \frac{2pr}{p + r}$$

5. References / Bibliography

Al Imran, A., 2021. *UCI Machine Learning Repository: Productivity Prediction of Garment Employees Data Set*. [online] Archive.ics.uci.edu.
Available at:
<https://archive.ics.uci.edu/ml/datasets/Productivity+Prediction+of+Garment+Employees>
[Accessed 2 September 2021].

Archive.ics.uci.edu. 2021. *UCI Machine Learning Repository*. [online]
Available at:
<https://archive.ics.uci.edu/ml/index.php>
[Accessed 2 September 2021].

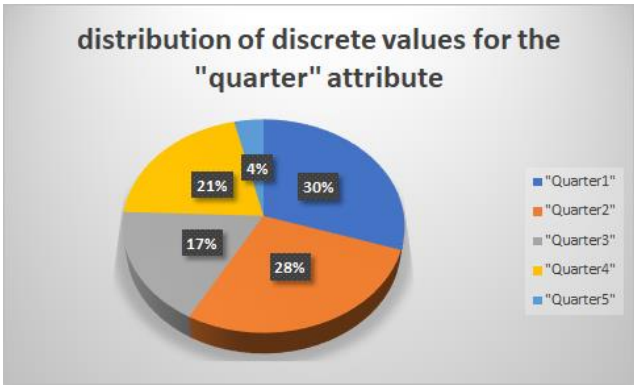
6. Appendix1: Data Investigation Report

1. Data Exploration

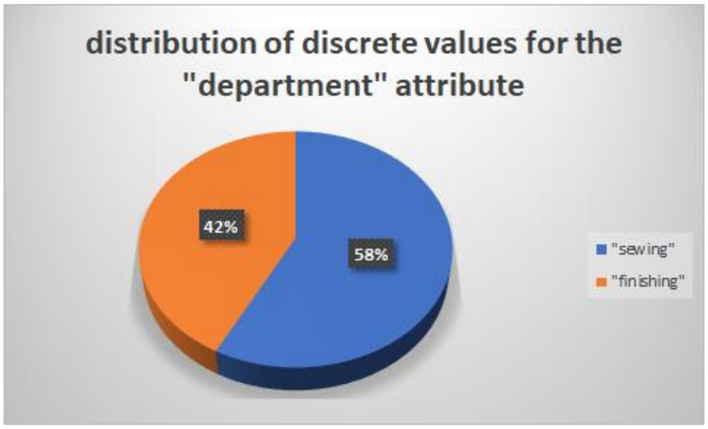
Data Attributes / Visualisations

date - a continuous interval-scaled data attribute which records the date on which the information was recorded.

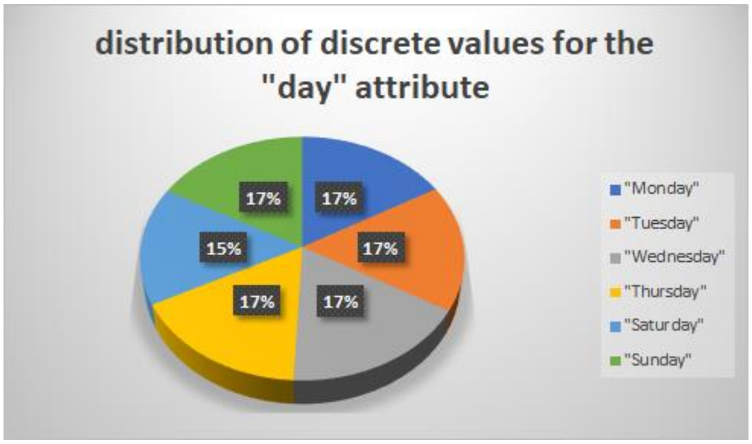
quarter - a discrete nominal data attribute which describes which ¼ of a given month in which this data was recorded. Below is a visualisation of the attributes distribution:



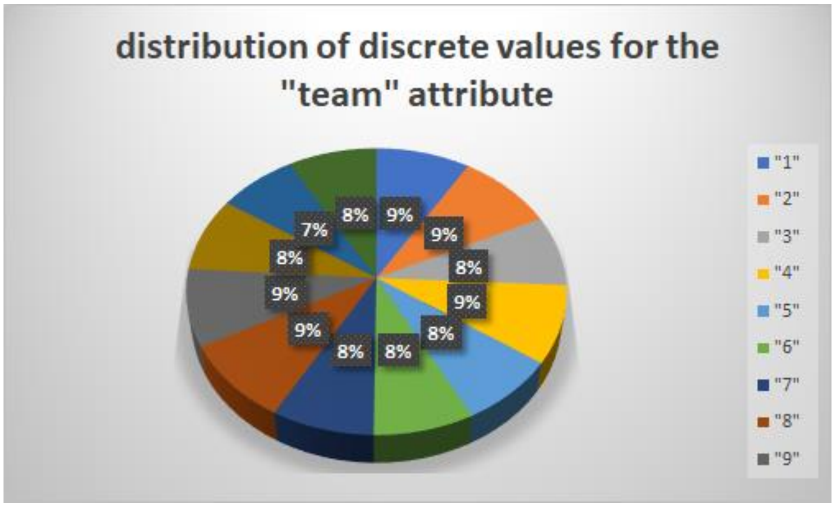
department - a discrete nominal data attribute which details which task the team has been assigned to in the given recorded instance. Below is a visualisation of the attributes distribution:



day - a discrete nominal data attribute which records what day the given instance was recorded on. Below is a visualisation of the attributes distribution:



team - a discrete nominal data attribute which records the id number of the team which the given instance refers to. Below is a visualisation of the attributes distribution:



targeted_productivity - a continuous ratio-scaled data attribute which details the productivity that the team was aiming for in the given instance.

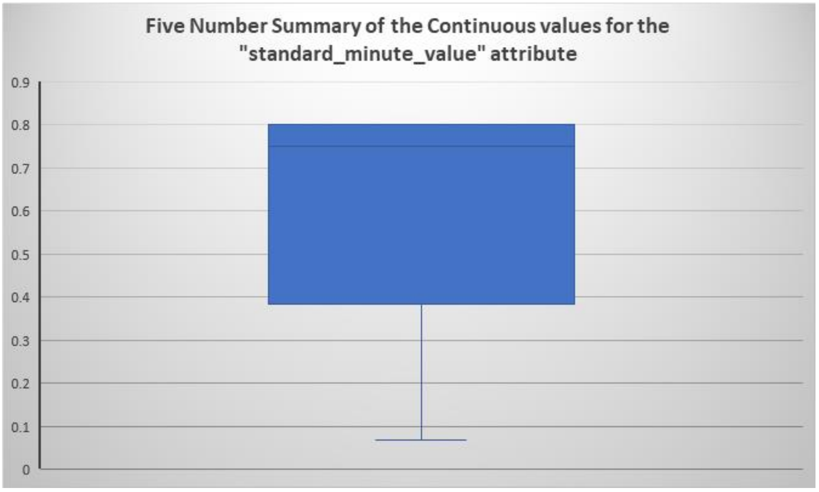
Min = 0.07

Q1 = 0.7

Median = 0.75

Q3 = 0.8

Max = 0.8



standard_minute_value - a continuous data attribute that represents the allocated time for a task

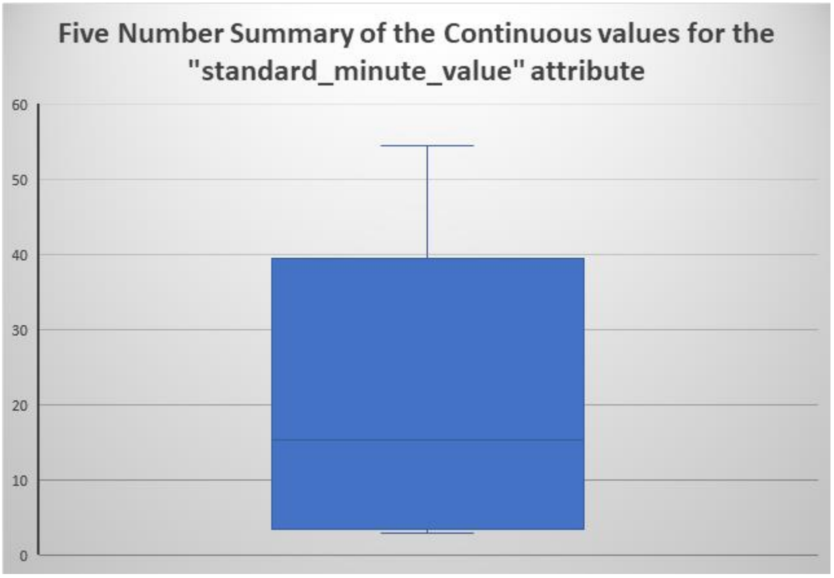
Min = 2.9

Q1= 3.94

Median = 15.26

Q3 = 24.26

Max = 54.56



work_in_progress - a continuous data attribute that details the amount of work currently in progress.

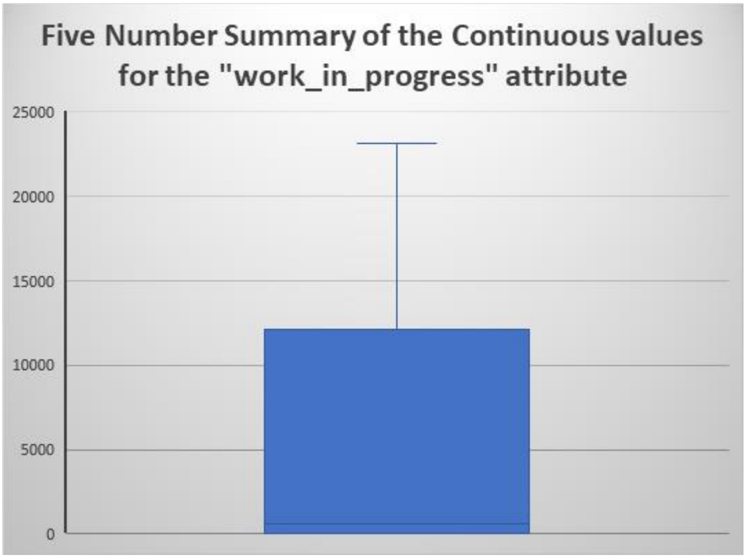
Min = 0

Q1 = 0

Median = 586

Q3 = 1083

Max = 23122



over_time - A continuous data attribute which records the amount of overtime by each team in minutes.

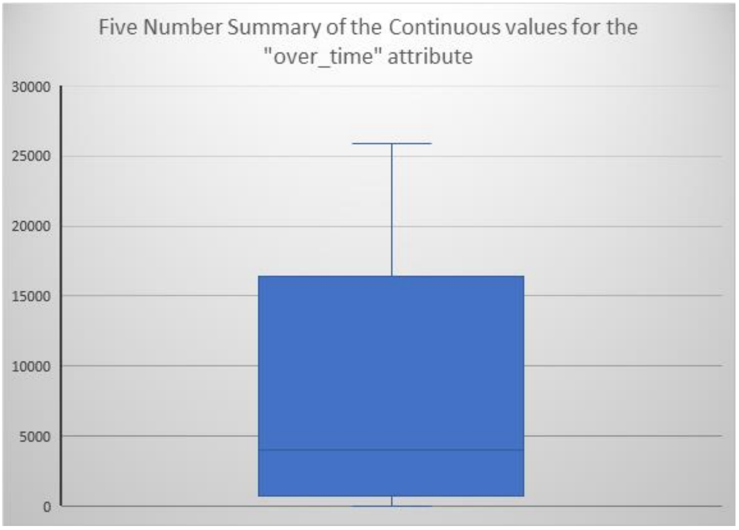
Min = 0

Q1 = 1440

Median = 3960

Q3 = 6960

Max = 25920



incentive - a continuous interval scaled attribute that represents the financial incentive (in Bangladeshi Taka) for a course of action.

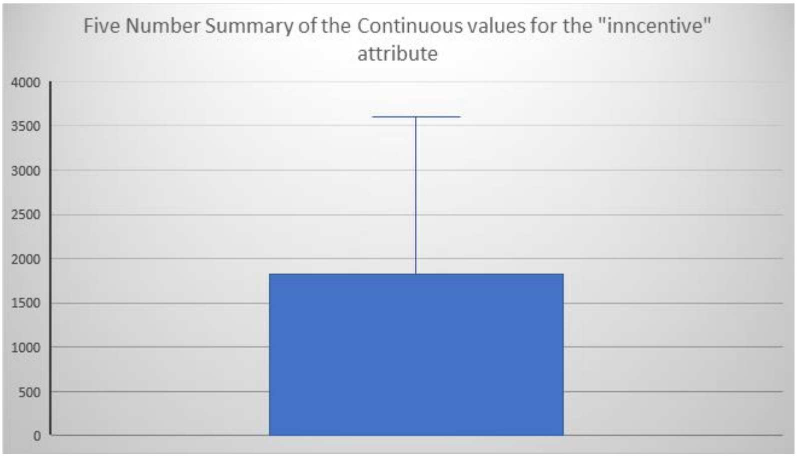
Min = 0

Q1 = 0

Median = 0

Q3 = 50

Max = 3600



idle_time - a continuous ratio-scaled data attribute that details the amount of times when the production was interrupted.

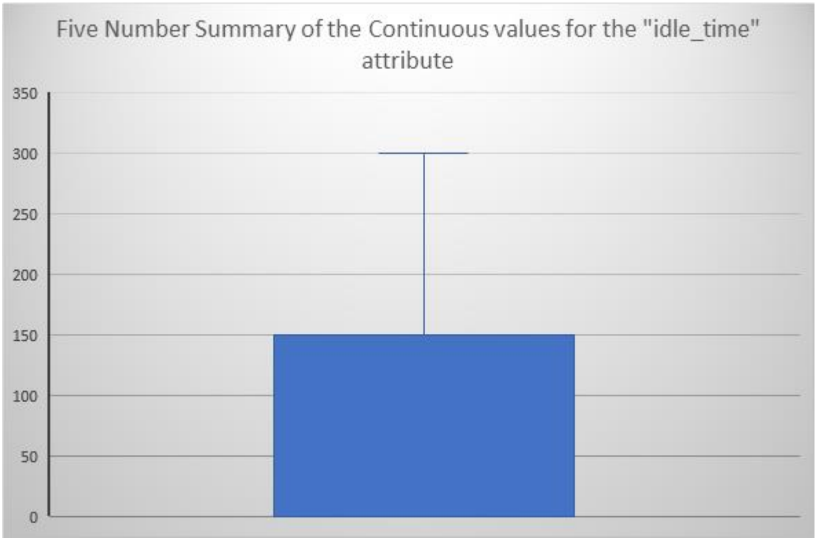
Min = 0

Q1 = 0

Median = 0

Q3 = 0

Max = 300



idle_men - an continuous interval-scaled data attribute that details the number of workers who were idle due to production interruption.

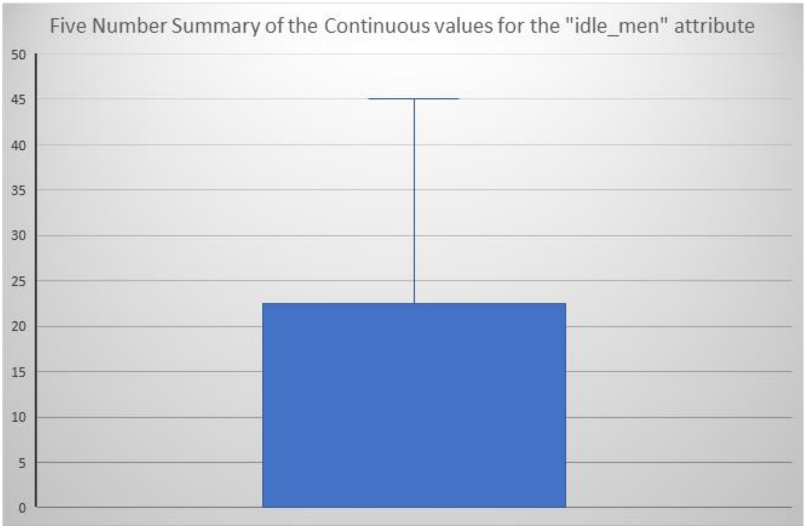
Min = 0

Q1 = 0

Median = 0

Q3 = 0

Max = 45



no_of_style_change - a continuous interval-scaled discrete data attribute which details the number of style changes in a particular product.

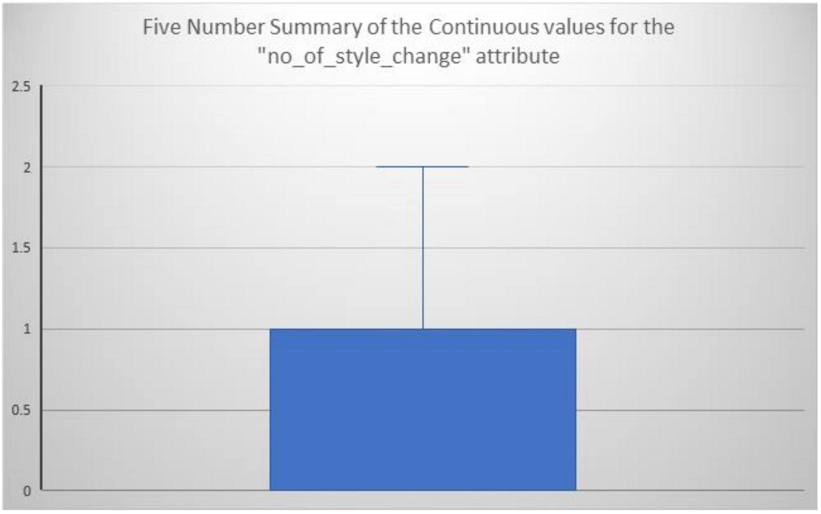
Min = 0

Q1 = 0

Median = 0

Q3 = 0

Max = 2



no_of_workers - a continuous interval-scaled data attribute which details the number of workers in each team.

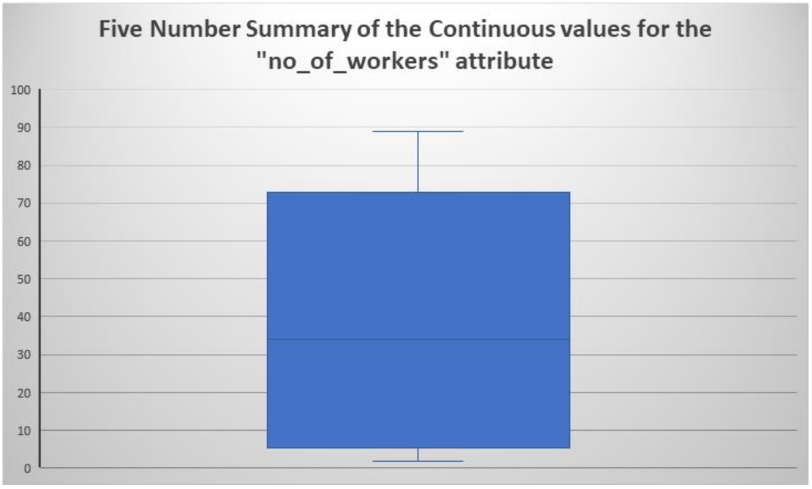
Min = 2

Q1 = 9

Median = 34

Q3 = 57

Max = 89



actual_productivity - a continuous ratio-scaled data attribute which details the productivity that the team achieved.

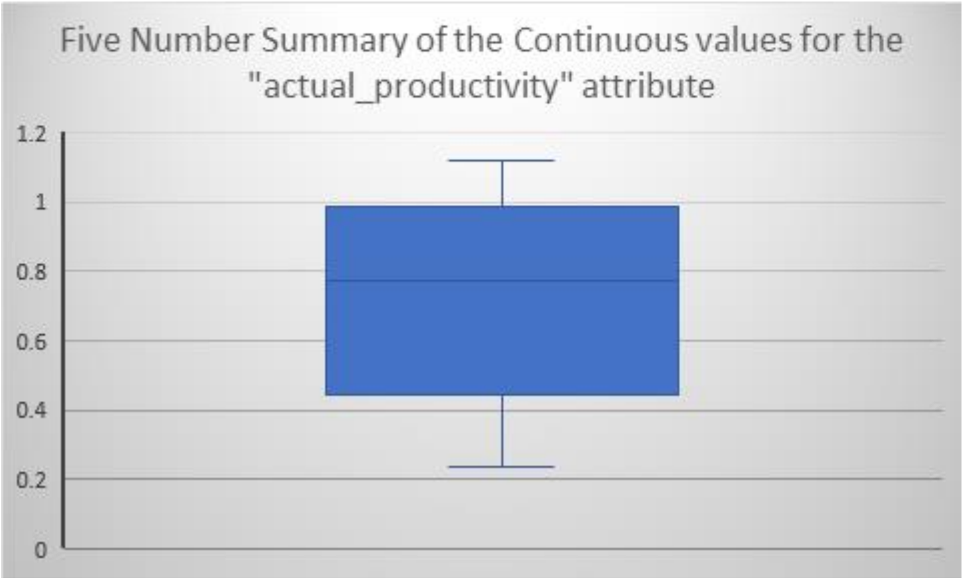
Min = 0.234 (three significant figures)

Q1 = 0.650

Median = 0.773

Q3 = 0.850

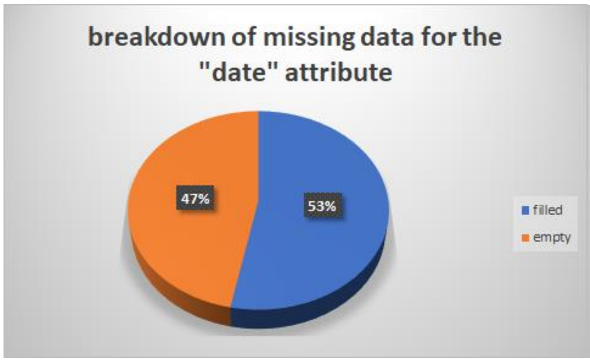
Max = 1.12

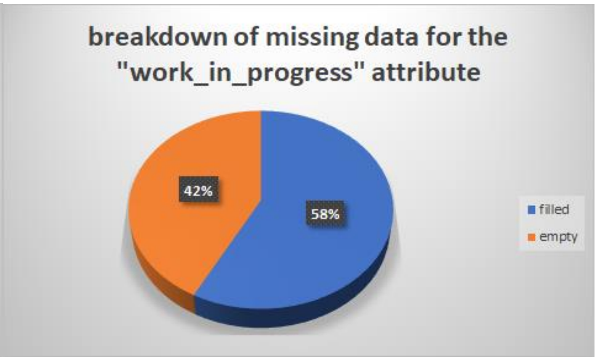


3. Data Pre-Processing

Below will be a list of different data quality concerns that we identified in the dataset.

Missing data - both the ‘date’ and ‘work_in_progress’ attributes contain missing data (as shown below by the pie graphs). In order to resolve this, default values will be input to fill in the gaps.





Actual_productivity - upon observing the ‘actual_productivity’ data attribute some noisy data instances were spotted which actually went over the possible maximum of 100%. In order to resolve this, all values over the maximum of 100% will be reduced to 100%.

Employee_count - upon observing the ‘employee_count’ data attribute some noisy data instances were spotted which weren’t whole numbers (which in this given context is not possible as you can’t have half a worker). In order to resolve this, all values with a decimal point will be rounded up.

Attribute binning - upon choosing the algorithms listed in the aims above it was discovered that in order to function that would require the binning of several continuous attributes. After deliberation it was decided that we should perform equal width binning to some of the attributes (such as the target and actual productivities).

7. Self-contribution

Sebastian

I wrote the data description for the data set in addition to creating a comprehensive excel data sheet in which many observations on the dataset were made. I then used the aforementioned excel spreadsheet to type all of the attributes, make the five number summaries, in addition to the various visualisations of the dataset shown within this document.