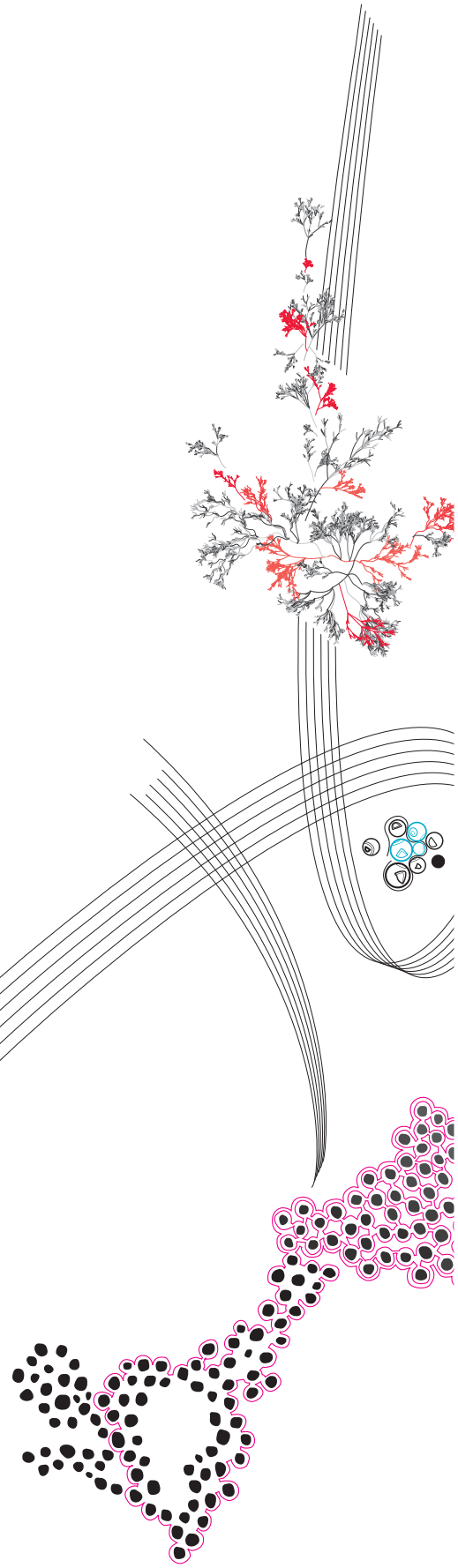

BSc Thesis Applied Mathematics

Machine Learning on Geothermal Heat Extraction

Cas Sitvast

Supervisor: Prof.dr.ir. B.J. Geurts

June, 2021

Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science

UNIVERSITY OF TWENTE.

Preface

I want to thank Bernard Geurts for his guidance and advise during the work on this bachelor's assignment. Without his feedback and helpful insights this paper would not have existed.
Also a great thanks to Ronald den Boogert for sharing his expertise and advise and for providing the practical context of the research.

Machine Learning on Geothermal Heat Extraction

Cas Sitvast*

June, 2021

Abstract

Drilling for oil and water has become a common thing, however still resources can be collected from old wells via the help of radial drilling. The way this drilling is done these days, does not always lead to a success. Therefore this bachelor's assignment will investigate collected data from drillings in order to illuminate the process and create more insight into factors that determine the progress. This is done with the help of an unsupervised machine learning algorithm called mean shift.
In this paper a tutorial will be described based on an online data set. An elaboration on how the mean shift algorithm works will also be given, and it will be applied to so-called jetting events that characterise successful steps in radial drilling data.
The drilling data as it was obtained, first needed to undergo a sequence of actions in order to obtain what we call patterns or features. These patterns were scaled to render a more uniform representation and subjected to the mean shift algorithm. This resulted in a separation of patterns, which was considered 90% accurate by an expert. An investigation was also done applying the algorithm to less uniform data. This was considered to have an accuracy of 94%. In both cases the algorithm was trained on all available patterns. The same investigations were also done with the algorithm trained on half of the patterns. This resulted in a mean accuracy of 69% on the completely uniform data and 62% on the relaxed uniform data. The data used in these results were obtained by one sensor, more research can be conducted in combining these results with signals obtained from other sensors. Research could also be done whether the combination of multiple sensors could result in accurate predictions of these patterns.

Keywords: unsupervised machine learning, drilling, data, mean shift

1 Introduction

For many years people have drilled for oil. They make a hole in the ground and pump the oil out until there is nothing left that can be extracted economically. However quite often there are still resources left in the ground, these resources are however not connected to the main hole and are too small to be profitable for a new hole. Therefore a new technique has been created, called radial drilling. This technique involves creating connections from the main hole to these small nearby pockets, significantly increasing the output of the main hole and thus raises the revenue of the drill site. [8]
During the making of these lateral holes, a lot can go wrong. And since the drilling takes place deep underground there is no easy way of telling whether everything is happening as planned and what measures one could take in case progress is slow.
This bachelor's assignment will investigate sensor data collected from a recent drill site. This will be done with a combination of data preparation, machine learning and expert

*Email: c.sitvast@student.utwente.nl

knowledge. Machine learning is a tool that can help with finding relationships in data that humans can not. This makes it a potentially very useful and effective tool, for the task at hand. Not only will it be useful for the current task, but it will also create opportunities for programs that use the results in real-time practice. The end goal of drilling companies will probably be such a program. For instance, such a program could be capable of analysing real-time data to determine in which state the process is, with what speed the drill is drilling and what can be done to improve the current process.
This end goal is, however, out of reach for this bachelor's assignment. But a start is made towards this goal. The goal of this assignment is to create so-called pattern profiles that belong to a state of the process. These profiles could be used in further research to real-time determine the state of a process. Finding these patterns is done with a combination of data preparation, machine learning and expert knowledge.

The organisation of the paper is as follows. What machine learning is will be explained in section 2. Section 3 is devoted to a tutorial for the use of machine learning. In section 4 the mean shift algorithm will be applied to real data and the obtained results are presented. The results drawn from these results are written in section 5. Finally in section 6 an outlook is given for further research.

2 Machine Learning

This section will discuss what machine learning is, the various 'types' of machine learning that are distinguished and briefly describe the implementation of some of the machine learning methods.

Machine learning is a technique founded in computer science in the late 1950s. In general, there are two types of machine learning: supervised and unsupervised. In supervised machine learning, the algorithms are trained on data that had some input from humans, whereas unsupervised machine learning algorithms are trained on data without any human input. This difference results in completely different approaches and results, but before going into more detail about the two different types a few definitions have to be made clear.

First of all, machine learning is often used to make sense of large amounts of data. What a machine learning algorithm often tries to achieve is to find an underlying relation in the data. In most cases, this is done by evaluating all the features of a data point.

A **data point** is one element out of all the data. Each data point usually has one or more features.

A **feature** is a characteristic or attribute that describes a data point.

Secondly a machine learning algorithm has to be **trained**. This training is the process in which the machine learning algorithm is learning. How this learning works in practice differs from technique to technique. It is common for machine learning to train the algorithm on only a part of the data. This subset of the data is often called the **training-set**, the remainder of the data is called the **test-set**. The test set is later used to check how accurately a machine learning algorithm performs.[2]

For example, suppose the data set is all existing cats and dogs. A data point would

then for example be a cat. The features could be something like the length of whiskers, height, the colour of hair, etc. And the goal of machine learning could be to determine whether a data point is a cat or a dog.

Supervised

As stated above, supervised machine learning is machine learning where the training data has had some human input. The training data is in this case already classified. In the example above this would mean that in the training data it is already specified whether a data point is a cat or a dog. Based on this training the supervised machine learning will then determine whether new data points are cats or dogs. How a supervised machine learning algorithm determines this is very dependent on the algorithm.

There are many different supervised machine learning algorithms, each having its benefits and downsides. The goal of this bachelor's assignment is, however, mostly focused on unsupervised machine learning Therefore the discussion on supervised machine learning algorithms is beyond the scope of this bachelor's assignment. However, since it would clarify how supervised machine learning works, an explanatory figure has been added, figure 1.
The K-Nearest-Neighbours (K-NN) algorithm was used to obtain this figure. This particular algorithm determines the class of a newly presented data point by evaluating its K nearest neighbours and applying the majority rule to determine the class the new data point should have.[5]

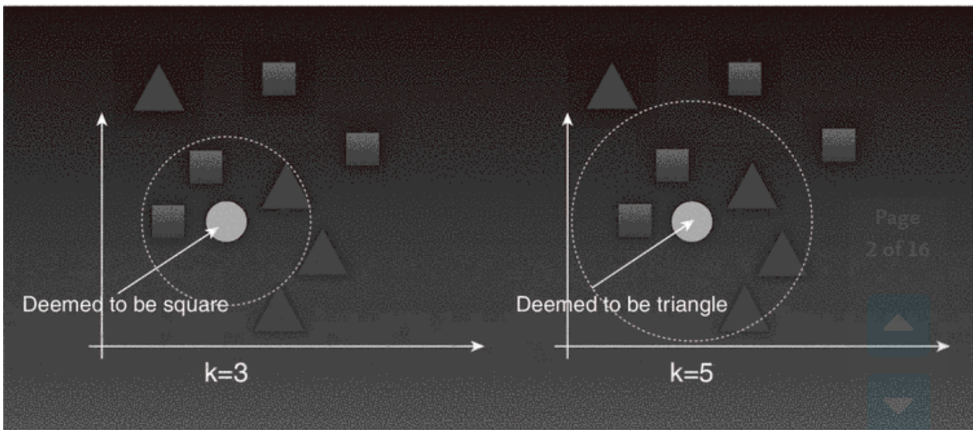

FIGURE 1: [5] K-Nearest-Neighbours algorithm example for k = 3, the new data point (circle) is deemed to be a square since 2 neighbours are squares, and k = 5 new data point is deemed to be a triangle, since three neighbours are triangles.

In the example in figure 1, the training data are all the squares and triangles already present in the picture. The question is to determine what the new data point (circle) should be. For k = 3 we see that the circle has two squares as closest neighbours and one triangle, so by the algorithm it should be a square. While in the other picture, for k = 5, the circle has three triangles as closest neighbours and 2 squares, so it should be a triangle.

Unsupervised

Unsupervised machine learning is where the training data has no human input. The learning of unsupervised machine learning is therefore different. Instead of trying to find the

relation between input and output, the algorithms often try to find similarities within the data. Depending on the used algorithm, it tries to cluster the data in different groups. Clustering is a type of technique that is used to classify a group of data points according to a specific routine.[4] Examples of these kinds of algorithms are mean shift, K-means or hierarchical clustering. This bachelor's thesis mostly uses the Mean Shift algorithm therefore a more detailed description will be given of this algorithm. The benefit of this algorithm is that it can independently estimate how many classes there are within the data. Which is something other algorithms can not.

Mean Shift

The Mean Shift algorithm aims to determine centre points in a data set, such that the density around these centres are as high as possible. In this case the number of classes is equal to the number of centre points. When the amount of centre points is given it will determine the optimal location for these centre points by first choosing a random location in the feature space and then moving the centre point in the direction of the region where the majority of the points resides. To determine this direction and proving that this algorithm converges is given by [1]. An illustration on how this algorithm works with 3 features is given in figure 2. In this bachelor thesis I am using the python library scikit-learn [7], that has this algorithm already programmed and in its database.

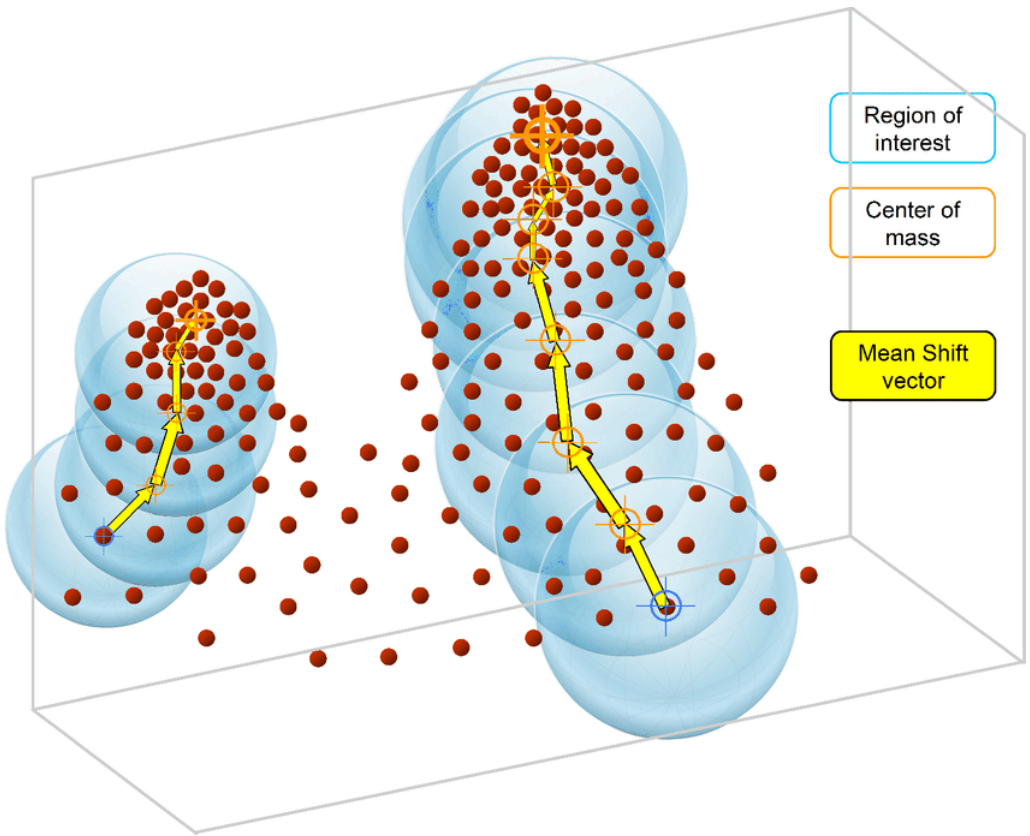

FIGURE 2: [6] Illustration of the mean shift algorithm finding two centres, yellow arrows indicate the direction that the centres of the mean shift algorithm move towards. The red dots are the data points.

3 Machine learning tutorial

To get familiar with the use of machine learning algorithms, a tutorial was considered based on an online data set which shares some basic properties with the drilling data that are considered later. The first subsection 3.1 shows what the available data for this tutorial contains and shows a visual representation of the data. In the second subsection 3.2 the results of the mean shift algorithms will be presented.

3.1 Data

The online data set used for this tutorial is called Iris [3] and can be found online for downloading, but is also standard in the scikit-learn library [7].

This data set is a collection of iris flowers from three different species: setosa, versicolor and virginica. The data set contains 150 flowers in total, 50 flowers of each species. The data set also contains information on these flowers, namely the sepal length, sepal width, petal length, petal width and to which species the flower belongs. These are what we called the features of the data points, mentioned in section 2.
If we make a plot for one feature in comparison to another, it will help in visualising the underlying correlation between features. If we number the features, so 1: sepal length, 2: sepal width etc. we can make a figure such that the first row and second column is the plot where sepal length (x-axis) is plotted against sepal width (y-axis). This results in figure 3. Note that the diagonal is left blank since this would be a feature plotted against itself and thus resulting in a diagonal line. Each colour represents a different species, red is the setosa, green the versicolor and blue the virginica.

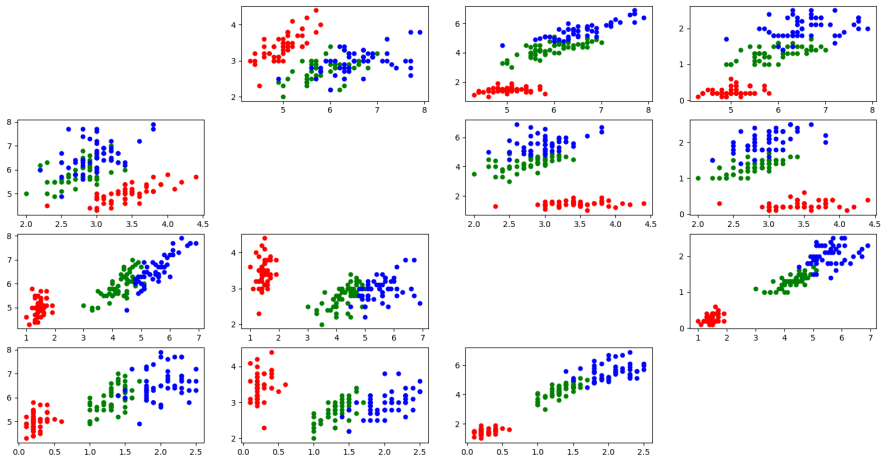

FIGURE 3: Every possible combination of features of the iris data set plotted. Each colour is a different iris species, red is setosa, green is versicolor and blue is virginica. This figure shows the clear similarity between the versicolor and virginica and the clear distinction of the setosa. What is on the axis is not important.

Figure 3 shows that the versicolor and the virginica are very alike and that the setosa is quite distinct. This means that upon applying the mean shift algorithm it is very likely

	Versicolor	Virginica	Setosa
Best class	Class 1	Class 1	Class 2
Positives	98%	100%	100%
Negatives	50%	51%	99%
False Positives	33.33%	32.67%	0.67%
False Negatives	0.67%	0%	0%
Total	66%	67.33%	99.33%

TABLE 1: Accuracy score of mean shift on all iris data

that the algorithm will find two classes instead of three. This happens since the algorithm finds the locations in which the most data points reside.

3.2 Results

Now that it is clear how the data is structured, a machine learning method can be applied. However, to know how well an algorithm works first, a measure of performance has to be defined. If nothing is known about the data it is hard to define how well an algorithm worked, but in this case, it is precisely known which data points belong together. This makes it easy to check exactly how well an algorithm performed. Upon checking this we are most commonly interested in five things for every class:

1. **Positives**: Percentage of data points correctly classified, e.g., an image of setosa is actually recognized as such.
2. **Negatives**: Percentage of data points correctly classified as not being of that class, e.g. an image of versicolor is recognised as not being a setosa (when considering the setosa class).
3. **False Positives**: Percentage of data points classified as being of that class, but are not, e.g. a virginica is recognised as a setosa.
4. **False Negatives**: Percentage of data points classified as not being of that class, but actually are, e.g. a setosa is recognised as a versicolor.
5. **Total accuracy**: Percentage of data points correctly classified

After applying the mean shift algorithm to the data, where the training set is all the data, we can see that the algorithm finds two classes, instead of three. This was likely to happen since, two of the iris species are highly related to each other, as can be seen in the feature plots 3. Assuming that each class belongs to one flower we can compute the statistics we are interested in. In table 1 the results are shown of the most likely species the class belongs to.

From the table we can see that the Mean shift algorithm clearly separates the Setosa species from the other two. It does not however, find the distinction between the other two types of flowers significant enough to create a separate class for them. What is interesting, is that when the algorithm finds the right class it is highly accurate in determining the members of that class (Class 1 $\rightarrow$ Setosa 99.33%).

Since the Versicolor and Virginca are considered to be one class by the algorithm, the percentage of positives is very high. Considerin the statistics for the Versicolor. Upon calculating whether all the versicolors are in this class we find that 98& of the versicolors are in the found class. However 50% of the flowers that are not supposed to be in the same

class are also in there. Which also explains the high amount of false positives. Since a third of the flowers are the correct flower, another third of the flowers are wrongly classified as the same flowers, and the last third is recognised as a different flower.
These are results of the Mean Shift algorithm when all data is used to train the algorithm. It can also be interesting to see what the results will be when the algorithm is trained with half of the data. The other half, the so called test data, will then be classified by determining which centre point, found bound the algorithm, is the closest. Which data points will be chosen for training is done randomly and the entire test is done a 1000 times. A histogram of the results of the total accuracy is shown in figure 4. In table 2 the means of these runs are shown.

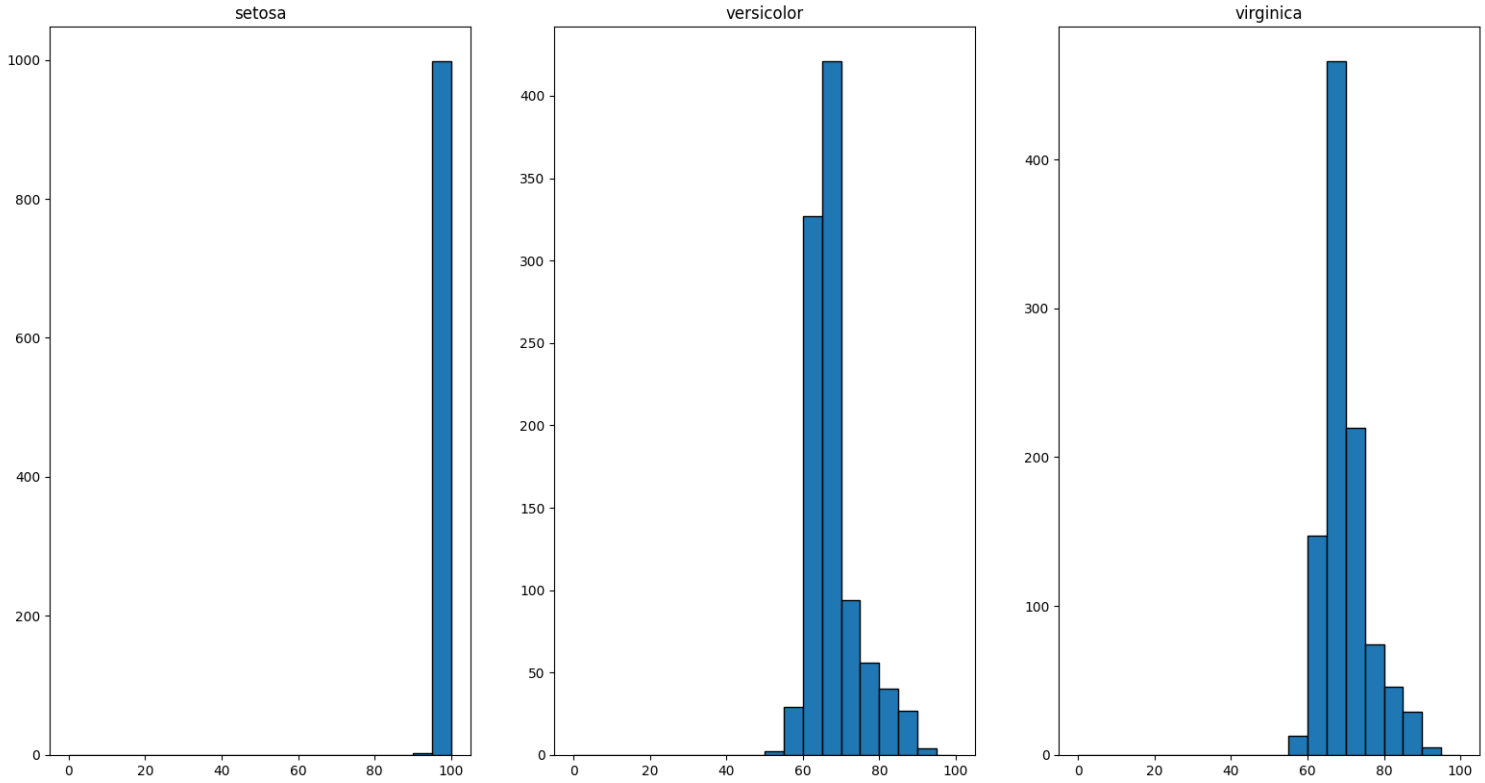

FIGURE 4: Total accuracy histogram of mean shift algorithm with random half as training set. 1000 runs in total.

With these results we can conclude that the mean shift algorithm applied in this way, is quite accurate when it comes to classifying Setosa. The other classes however, are probably too much alike for the algorithm to find a distinction. Also a variation in the training data could result in drastically different accuracies.
The algorithm finds the most dense location in the data and calls that the centre of a class, which means that if more data points are added this could create a denser spot in the data, thus creating a clearer image and possibly a better classification.

	Setosa	Versicolor	Virginica
Positives	100%	96.16%	92.69%
Negatives	98.61%	54.44%	59.03%
False Positives	0.92%	30.55%	27.39%
False Negatives	0%	1.34%	2.63%
Total	99.08%	68.11%	69.98%
Min Accuracy	94.67%	56%	56%
Max Accuracy	100%	93.33%	92%

TABLE 2: Mean accuracy score per iris species of 1000 runs, random half of data as training set

4 Machine learning on data set Radial Drilling

With the experience obtained from the tutorial, described in section 3, we can advance to the data of interest. In subsection 4.1 the visualisation and preparation of the data at hand will be discussed. In subsection 4.2 the application and results will be presented.

4.1 Data

For this bachelor's assignment, data was provided by RadialDrilling[8] a company specialised in radial drillings, they have done several of these drillings and lent us the sensor data of one these drillings. Even though this data contains the information of several sensors and several drilling processes, only one sensor data of one process was investigated in this bachelor's assignment. This was to keep it simple, find out whether machine learning is useful in this context, and to create a baseline upon which further research can be conducted. Since this data is very valuable it was lent under a nondisclosure agreement. Therefore figures of the data and descriptions of sensors have been distorted in order to uphold this agreement.

With this in mind, we will refer to the investigated sensor data as the Process data for simplicity. A plot of the Process data is shown in figure 5.

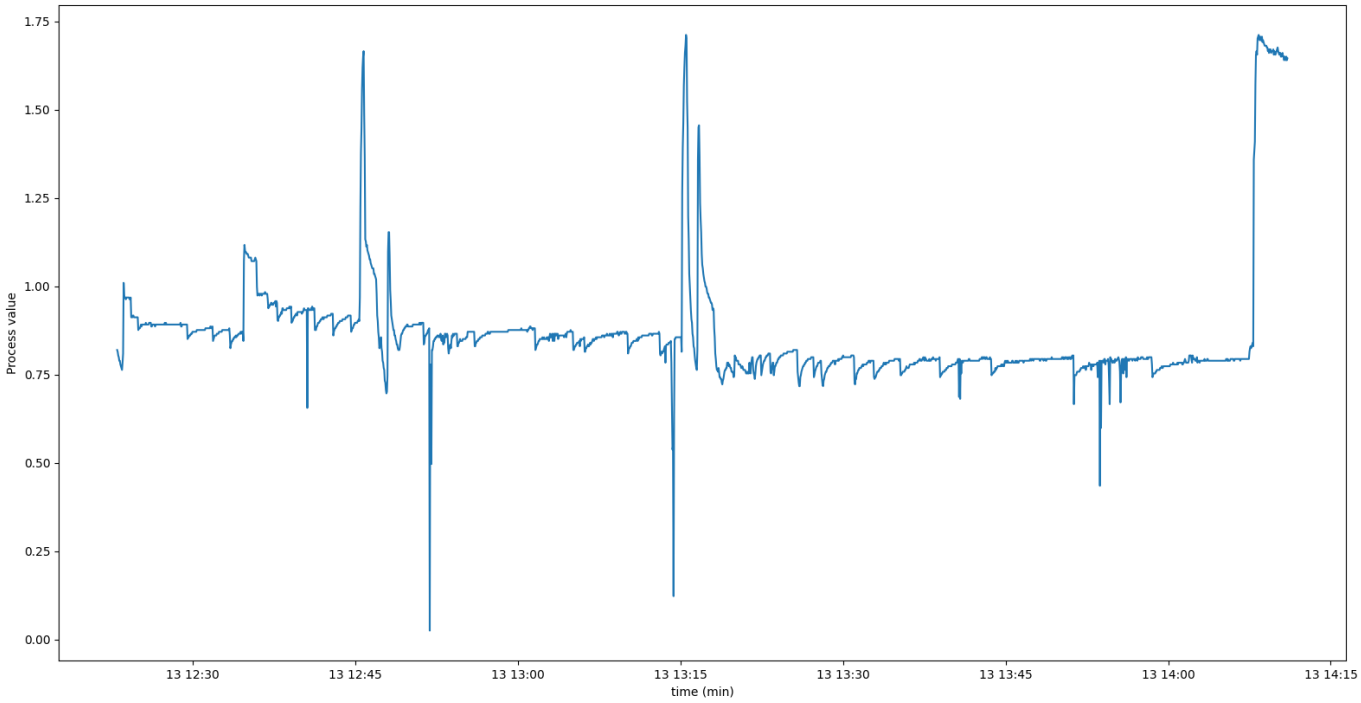

FIGURE 5: Raw Process data

When we take a closer look at figure 5, we can already see some patterns in the data. It seems as if there are three events illustrated by this particular data stream:

1. The values follow some sort of arc function and then drops down.
2. The value increases rapidly, then decreases rapidly
3. The value decreases rapidly and then increases rapidly

After consultation with an expert it was determined that the process is doing fine in events of type 1, the operators at the drilling site were causing events of type 2 and events of type 3 are unwanted behaviour. The ultimate goal would be to real time evaluate the sensors and determine in which state of the drilling process is currently achieved. However, as stated before this is beyond the scope of this bachelor's assignment. Instead we look at a small part of the complete process and determine whether this process is working correctly. To achieve this the first step is to separate the data. This could be done by simply looking at the time of every event and noting that down. But that would be a very tedious job and, more importantly, it would not be scale-able, so instead the data was separated by stepping through every data point and subjecting it to a certain rule. After the first separation part you step through the newly found separations and subject those to a new rule, and continuing on like this until a suitable separation is found. What is considered suitable is again done by an expert.
If we let X represent all available Process data, then $X[i]$ is the i^{th} data point for $0 < i <= N$, where N is the total number of available data points. The first separation rule is based on the events of type 2. We decide to split when for some sufficiently

large α, $\alpha + X[i] > X[i-1]$ or $X[i] < X[i-1] - \alpha$ for $1 < i <= N$. This α should be larger then the maximum distance between two consecutive points in the desired pattern. After testing multiple of these values the best result was found using $\alpha = 0.075$. This results in figure 6, in this figure every colour is a set of data points considered to belong together. The split happened at the point where one colour changes into another.

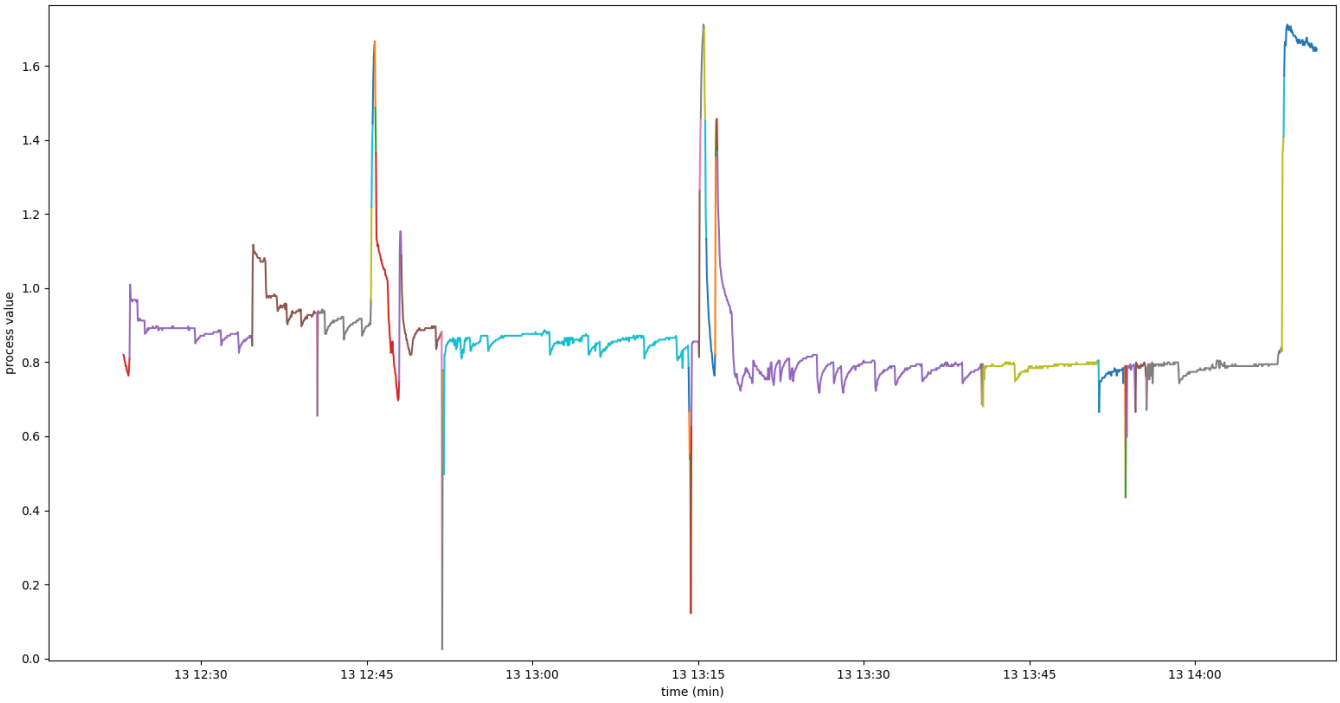

FIGURE 6: First split on Process data based on rapid increase followed by rapid decrease. The split happens at every point where the colour changes.

By expert judgement we know that the value of the Process should never be higher than 1.375 and assuming that data sets containing only a few points do not contribute anything, these data sets can then be removed from consideration. The result is shown in figure 7.

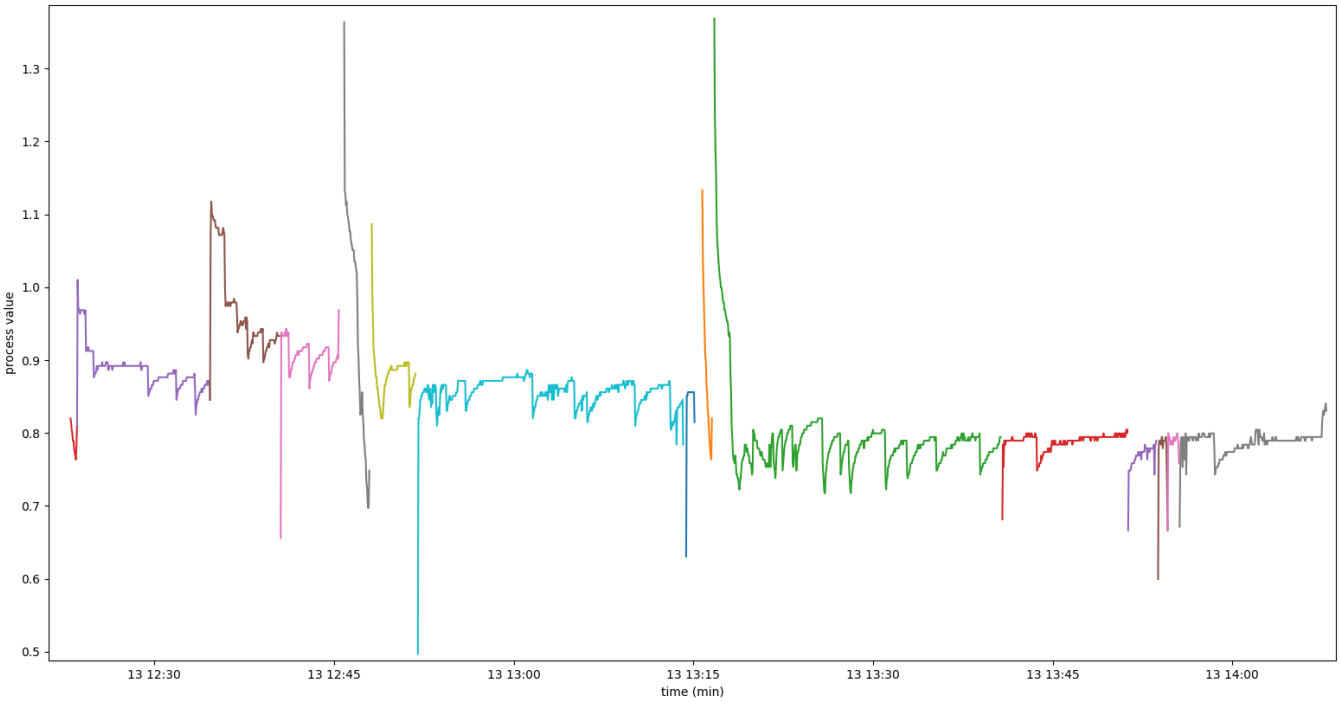

FIGURE 7: Process data after removal of data sets containing a few points or having a value exceeding a certain limit

Lastly a separation is done based on events of type of 3. The separation will be at the first point that goes up after we continuously went down. This results in 48 data sets which will be called Patterns. All the found patterns are shown in figure 9 and for reference the final separation is shown in 8. We call this set of 48 patterns P and denote the value of a pattern $p \in P$ at index i to be $p(i)$.

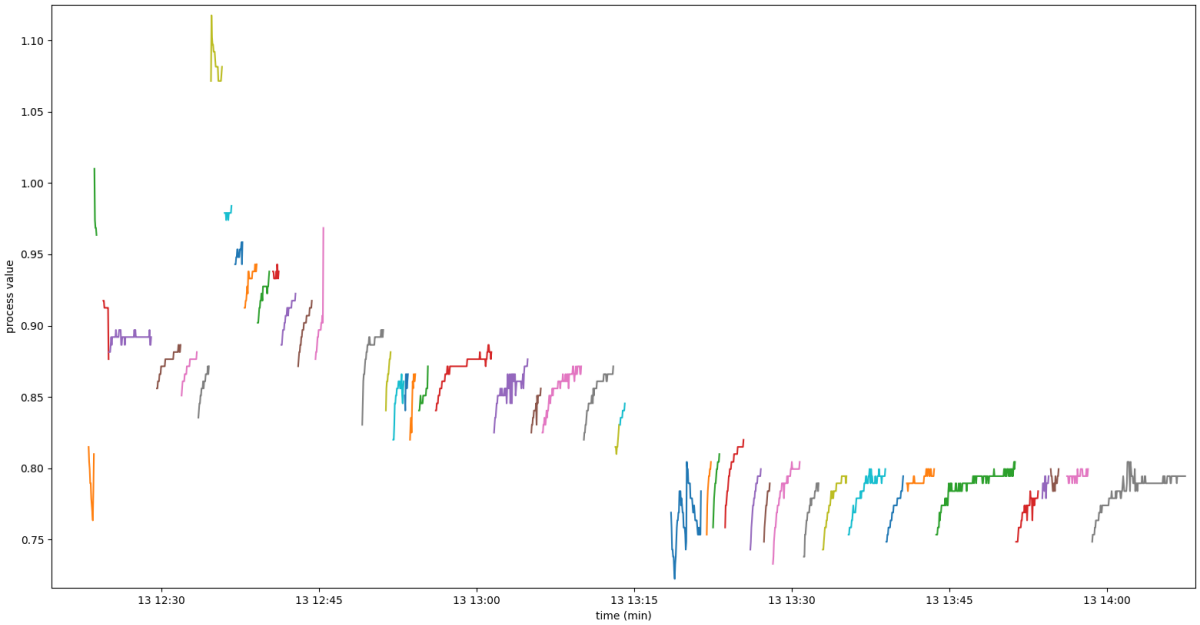

FIGURE 8: Process data after splitting at points that go up after continuously going down.

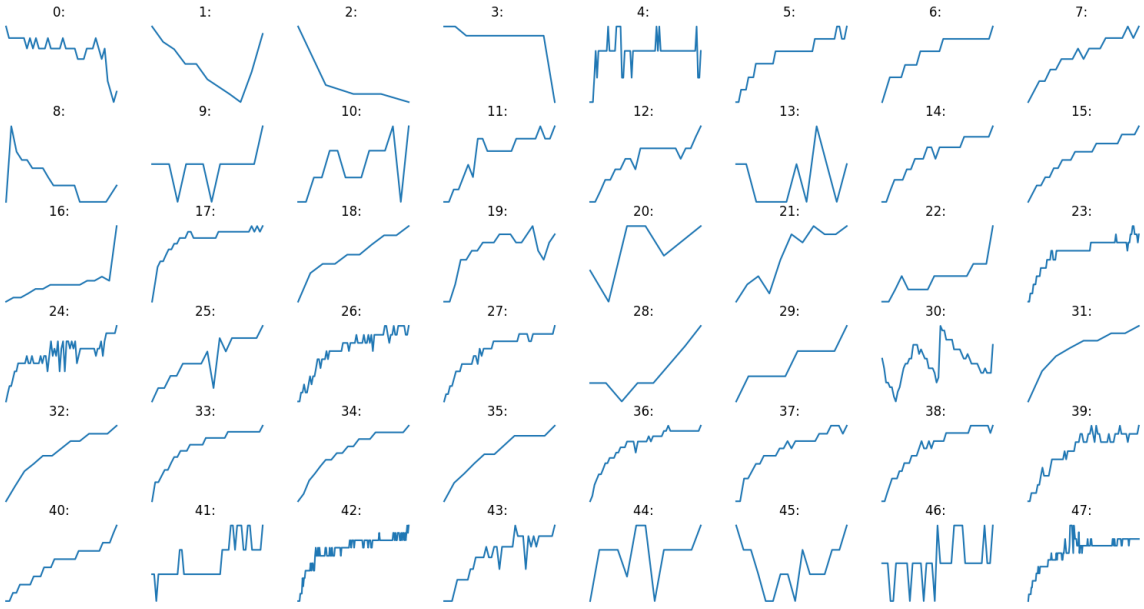

FIGURE 9: All patterns after sequence of separation rules. Automatically scaled by python to fit the same size box. So x and y-axis are different in every sub figure.

Figure 9 is a plot of every pattern in which python has automatically scaled every pattern to fit the same size box. Doing this immediately shows similarities between some of the patterns, therefore uniforming the patterns might be a useful first approach. Uniforming the data breaks down into two problems: uniforming the length of the pattern and uniforming the height of the pattern. However in order for the mean shift algorithm to work, it is also needed for every pattern to have the same amount of data points. This is needed, because in this case every pattern is an element of interest and its data points are the features of that element. To compare it to the tutorial of section 3. Every pattern is a flower and every data point tells us something about that flower. It is possible to plot all the features against each other, just like in the tutorial. This would result however, in a 200 by 200 figure and nobody would be any wiser, therefore it is left out.
There is a problem to this though. Because in the preparation of the data it is made sure that there are no patterns with less than ten data points. There is however, no restriction on the maximum length of data points. This causes that some patterns have more data points than others. In order for the mean shift algorithm to work, it is required that every pattern has the same amount of data points. If this would not be the case the algorithm would try to compare a vector in R^n with a vector in R^m $(n \neq m)$ which would be nonsense.
So in order to solve this length problem we opt to create a new pattern set called $P^\star$ in which every pattern is created from a pattern in P and for which $D_n(i) = D_n(j)$ for all $i, j \in P$. In which $D_n(i)$ is the amount of data points in pattern i. Also there has to be a one-to-one transformation T that links the indices from pattern $p \in P$ to its new pattern $p^\star \in P^\star$. This transformation T together with the following other requirements have to be met. In this list we have $p \in P$, $p^\star \in P^\star$ and i, j to be any two data points in p.

1. $p^\star(0) = p(0)$
2. $p^\star(T(i)) \geq p^\star(T(j))$ whenever $p(i) \geq p(j)$
3. $p^\star(T(i)) \leq p^\star(T(j))$ whenever $p(i) \leq p(j)$

If any of these would not hold the new pattern could be totally different compared to its original one.
Another rule has to be created since these rules do not solve the issue of having too few points within the pattern. A solution again is found in the python plots. These plots connect every data point with a straight line. So as a first try extra points can be added in between existing data points that have as value, the value that lies on the straight line connecting two data points. Using all these rules and this idea of adding points in between, we create algorithm 1. This algorithm then creates this set $P^\star$

Algorithm 1 Uniform Length

1: maxLength := max(# data points in pattern 1, #data points in pattern 2, $\cdots$)
2: newPatterns = Empty
3: **for** Pattern $\in$ All Patterns **do**
4: D_n := # data points in Pattern
5: NewPattern = EmptyPattern
6: **if** $D_n <$ maxLength **then**
7: index := 0
8: **while** index $<$ maxLength **do**
9: **if** index $\in$ $rounded(\frac{maxLength}{D_n-1} \cdot(0, 1, \cdots, D_n - 1))$ **then**
10: newPattern at index = value of Pattern at $\frac{index}{D_n-1}$
11: **else**
12: a = index - index of prev value in pattern
13: newPattern at index = prev value in Pattern + a*(prev value in Pattern + next value in Pattern) / (# missing values in between + 1)
14: **end if**
15: index + 1
16: **end while**
17: **else**
18: newPattern = Pattern
19: **end if**
20: add NewPattern to NewPatterns
21: **end for**
22: **return** newPatterns

For a more clearer view on how this works an example is added. In this example we consider pattern $p \in P$ for which $D_n(p) = 10$.

Example 1. Suppose max Length = 20, then $\frac{maxLength}{D_n-1} \cdot(0, 1, \cdots, D_n)$ of line 7 in the algorithm will be the set $\{0, 2, 4, 7, \cdots, 19\}$. We will create a new pattern, which has the same initial and end point and goes through the same points as the original pattern in the same order, but we add more points in between. So in this case if we let the new pattern be $p^\star$ then, $p^\star(0) = p(0)$, $p^\star(1) = p(0) + 1 * \frac{p(0)+p(1)}{2}$, $p^\star(2) = p(1)$, $\cdots$, $p^\star(5) = p(2) + 1 * \frac{p(2)+p(3)}{3}$,$p^\star(6) = p(2) + 2 * \frac{p(2)+p(3)}{3}$,$\cdots$, $p^\star(19) = p(9)$.

After this algorithm we have a set of patterns which all have the same number of data points, hence if we take the length between every data point to be equal the new pattern set $P^\star$ is uniform in length.

To uniform the height of every pattern we choose a range in which all the patterns have to be, for simplicity we choose the range [0, 1]. Then for every pattern we must set $\min(P) = 0$ and $\max(P) = 1$ and scale every other value accordingly. This can be done with algorithm 2, the case $\min_h = \max_h$ does not happen, since this would imply a straight line.

Algorithm 2 Uniform Height

for p ∈ P **do**
    $\max_h$:= max(values of p)
    $\min_h$:= min(values of p)
    for i such that $0 \leq i < D_n(p)$ **do**
        set p(i) := $\frac{p(i)-\min_h}{\max_h - \min_h}$
    end for
end for

Applying algorithm 2 after algorithm 1 to our set P, we obtain P^u in which every value is between 0 and 1 and every pattern has the same length. Plotting P^u results in figure 10. Clearly some overlapping pattern are visible, but there are also wrong patterns still in there. Sorting the good patterns from the bad patterns is tried with the Mean Shift algorithm.

FIGURE 10: All uniformed patterns of Process data plotted in overlap

4.2 Mean shift on Process data

Uniformed Proces data

Now that the data is transformed to the set P^u the mean shift algorithm can be applied. First of all we let the training data be all available patterns in P^u. When applied, the Mean Shift algorithm finds 12 different pattern groups shown in figure 11. In order to get any reference on how well the algorithm performed, an expert looked at the patterns in figure 9 in order to determine a set of patterns that was according to him good enough. We call this set P_e and contains the patterns: 5, 6, 7, 11, 12, 14, 15, 17, 18, 19, 21, 23, 24, 25, 26, 27, 29, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43 (the numbers come from the figure). With this an accuracy test can be done on every pattern group found by the Mean

Shift algorithm and we can classify the group with the highest score to be the group we are looking for. In table 3 the results of the group with the highest accuracy are shown. The figure shows that the algorithm found 12 pattern groups. This could be reasonable since, a lot is still unknown about the process and every pattern group could be a representation of an unknown event happening. Group 2 could, for instance, be the pattern associated with hitting a large rock. Many of these classes are however, represented by only one pattern. This does not provide a firm basis for this idea. It is however, not of great importance to the goal of this bachelor's assignment. Since we opt to classify into two groups, the patterns in P_e and everything else. The table 3 shows that it can do this job with an accuracy of 90%. The mean shift algorithm clearly finds 93% of the patterns in P_e to be of the same class. Determines that 84% of the patterns that are not in P_e, indeed do not belong in P_e. 6% of all patterns are wrongly classified to be in P_e that actually do not belong there and 4% are wrongly classified to not be in P_e even though they should be.

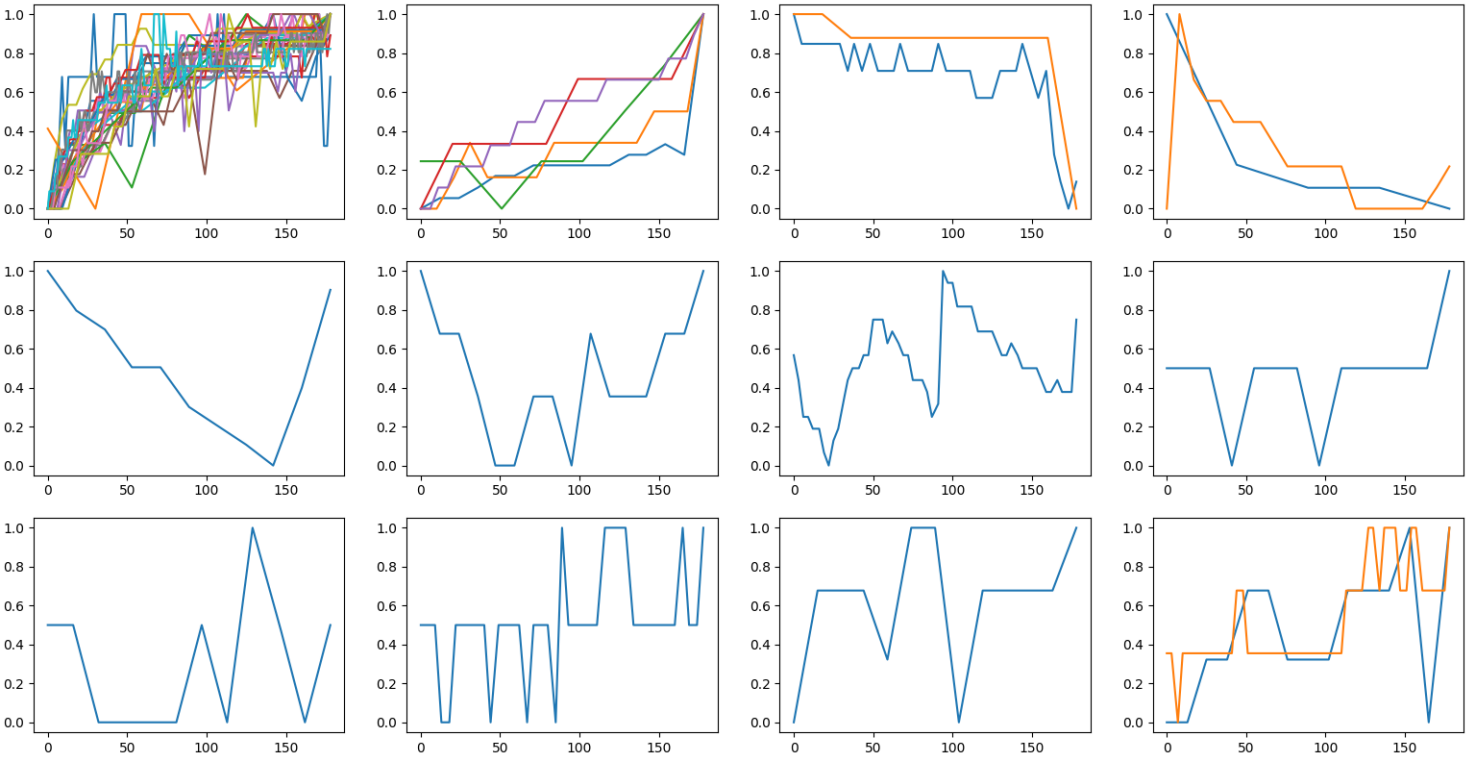

FIGURE 11: Pattern groups found by Mean Shift algorithm on Process data after uniforming.

	Random train on half
Positives	70.67%
Negatives	71.25%
False Positives	21.04%
False Negatives	14.24%
Total	69.18%
Minimum Accuracy	45.83%
Maximum Accuracy	88%

TABLE 4: Accuracy results of mean shift algorithm with random half of data as training set.

	Means Shift
Positives	93%
Negatives	84%
False Positives	6%
False Negatives	4%
Total	90%

TABLE 3: Accuracy results of mean shift algorithm applied to uniformed patterns

To obtain these results all the different patterns were used to train the model. To see how well the algorithm performs when only half of the patterns are given we do the same kind of tests as in the tutorial. So we do a 1000 runs in which half of the patterns are used to train the algorithm and the other half is used for validation. The results are shown in table 4 and a histogram of the results are shown in 12.

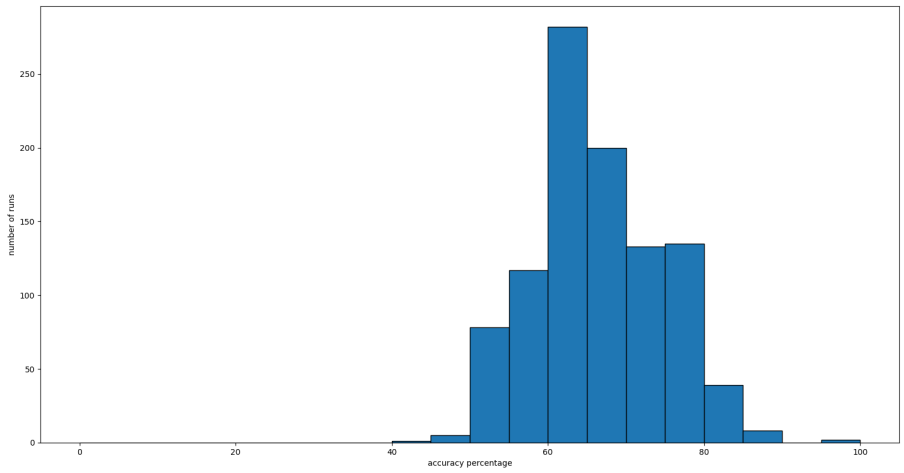

FIGURE 12: Histogram of Mean shift algorithm on 1000 runs with random half as training set.

	Relaxed height train on all	Random train on half
Positives	96.55%	55.03%
Negatives	89.47%	86.58%
False Positives	4.17%	12.76%
False Negatives	2.08%	24.87%
Total	93.75%	62.36%
Minimum Accuracy		45.83%
Maximum Accuracy		88%

TABLE 5: Accuracy scores of mean shift algorithm on Process data with relaxed uniforming. In the second column the algorithm is trained on all of the patterns. In the third column the average of 1000 runs is shown, where the algorithm is trained on half of the data

Process data with relaxed uniforming

After the decent results of the Mean Shift algorithm on the uniformed data it is interesting to investigate whether a relaxation of the uniforming can be done and still get decent results. Starting with a relaxation of uniforming the height. For this the values will not be scaled anymore to be between 0 and 1, but will be moved in order for the patterns to start from the origin. This is necessary because otherwise it would matter at what time the pattern began. This time the algorithm finds four groups, shown in figure 13. The accuracy score can be seen in table 5. In this table also the result from a 1000 runs with a random half as training set is included and the histogram of these 1000 runs is shown in figure 14.

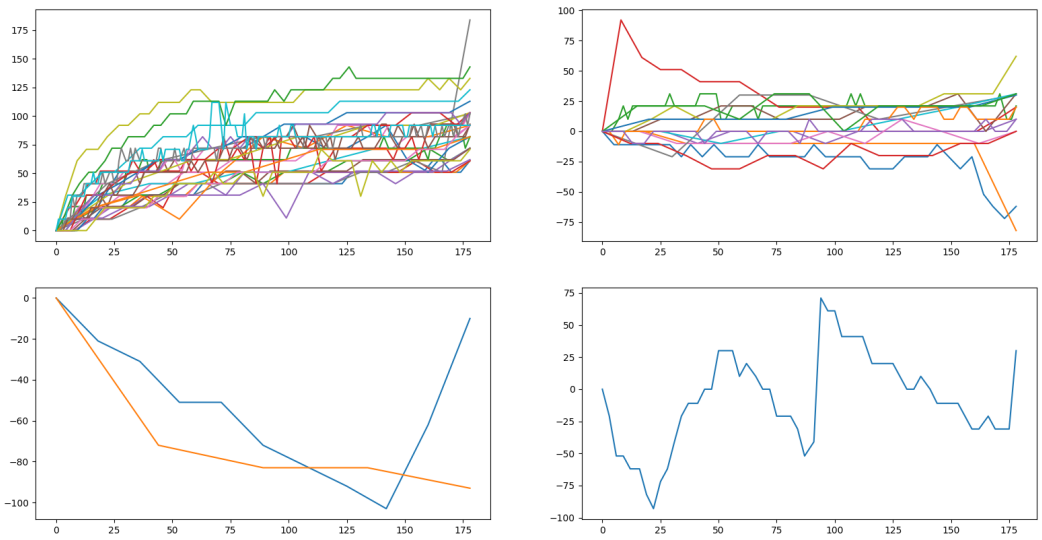

FIGURE 13: Pattern groups of Process data with relaxed uniforming, found by the mean shift algorithm

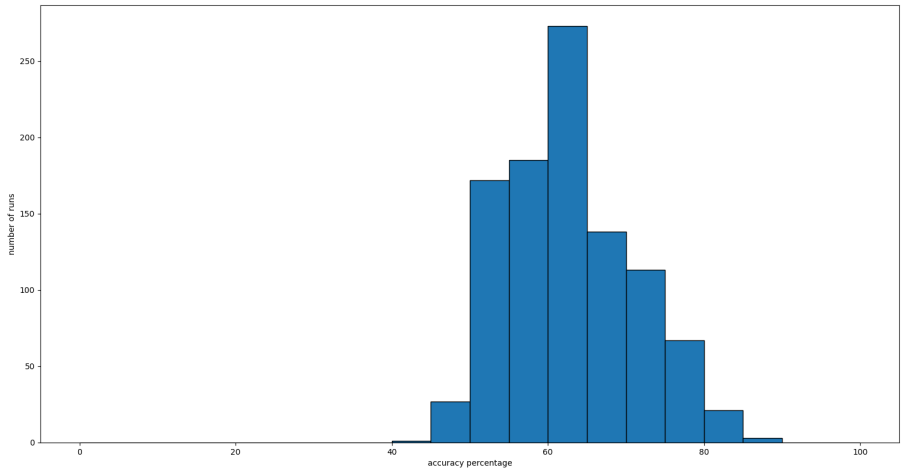

FIGURE 14: Histogram of Accuracy score of mean shift algorithm, trained on random half of Process data

5 Conclusion

This section will discuss the results obtained in section 4.

Looking at the results of table 3 it is clear to see that the mean shift algorithm does a good job in separating the good patterns from the wrong ones. Even though the algorithm makes some mistake in most cases it will classify the pattern in the correct group. Of course there are only two groups in this bachelor's assignment, which could make it easier for the algorithm. If more pattern groups are allowed the algorithm could also be of use, given that the other groups are sufficiently distinct. For otherwise the same thing will happen as in the tutorial of section 3, in which two plants are too much alike.

What can be concluded from table 4 is that the amount of data and the quality of data is very important. Since the accuracy of the algorithm can apparently range between 45% and 88% depending on which patterns where chosen for training. This is quite a large range and immediately questions the viability of the algorithm. Since an accuracy of less than 50% is worse than just guessing by flipping a coin, since this yields an accuracy of 50% on average.

What is interesting to see, is the difference in accuracy between table 3 and table 5. In both cases the algorithm was trained on all patterns, yet upon relaxation of the uniforming of the height the accuracy is better. Which was contradictory to the hypothesis that uniforming the data would yield better results.

6 Discussion

This section will discuss the used approach and investigate opportunities for further research.

During the preparation of the data (section 4.1), the data was uniformed according to a self-made algorithm. There are different ways of uniforming data, so it would be interesting to investigate whether the algorithm still works when the is scaled according to a different uniformisation. Or that a different uniformisation could create better results.
Due to time constraints it was not investigated whether the patterns found in this paper, could also have a clear pattern in the space of some other sensor. Or whether data of other sensors could have improved the distinction of patterns concerning the sensor used in this paper. The expert, often advised during this paper, expects just that. This could very well be the next step on the road to the end goal described in the introduction (section 1). Another path to this goal, could be further research into the patterns of figure 2 in order to see whether events that are known to happen sometimes (like the drill head being too blunt to drill), could be linked to one of these patterns. Or when more data is collected whether these patterns return, more often.

References

[1] Dorin Comaniciu and Peter Meer. "Mean Shift: A Robust Approach toward Feature Space Analysis". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2002), pp. 603–619.

[2] Vivien Deparday et al. *Machine Learning for Disaster Risk Management*. Washington, D.C. : World Bank Group, 2019.

[3] *Iris flower set*. URL: `https://en.wikipedia.org/wiki/Iris_flower_data_set`.

[4] Matthew Kyan et al. *Unsupervised Learning, A Dynamic Approach*. IEEE PRESS, IEEE-Wiley, 2014.

[5] Wei-Meng Lee. John Wiley Sons, 2019, p. 205. ISBN: 978-1-119-54563-7. URL: `https://app.knovel.com/hotlink/toc/id:kpPML0000S/python-machine-learning/python-machine-learning`.

[6] *Mean Shift picture*. URL: `https://www.researchgate.net/figure/Intuitive-description-of-the-mean-shift-procedure-find-the-densest-regions-in-the_fig3_326242239`.

[7] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[8] *Radial Drilling*. URL: `https://radialdrilling.com`.