CS5014 - Machine Learning

Practical 2 - Classification of objects using a radar signal and machine learning

Introduction

This practical introduces a binary and multiclass classification task using real world data. In both cases the data provided was collected via 4 back to back channels containing 64 components each and it recorded the readings of different objects. The maximum, minimum and mean signal values were recorded for each channel which resulted in 768 features being registered.

The binary classification consisted of identifying if the item detected was a book or a plastic case whereas the multiclass task classification added three additional classes to identify (air, knife and plastic case). This assignment was divided into various steps and the findings of the assignments are discuss bellow. For your reference, Scikit-learn is referred in this report as Sklearn.

Process

Data loading, cleansing and splitting

The data was initially loaded into Pandas data frames and the labels were loaded from the Json type files (Note these files was modified to follow the Json Format). The data frames were checked for invalid entries (null values) however, all entries were complete and imputing/cleaning the data was not required. The data entries loaded contained labelled and unlabeled data.

The labeled datasets were split into a training and test set (30% testing) straight away after loading to prevent data leakage. This test date will later be used to evaluate the models developed. Since the dataset was small in size, it was decided a larger segment of the data should be used to train the model. Furthermore, the split was randomized and stratified to prevent inducing a bias in the training of the model through due to an uneven distribution of the data. This step was common for both classification tasks, however the report will now focus on each task separately.

Binary classification task

Plotting and analyzing the data

The data was plotted in different ways to observe certain patterns and to have a general overview of the signal data. To gain insight into the patterns for each feature, all the data was separated according to its classification label and plotted as seen below in Figure 1 where the Y-axis represents the signal amplitude.

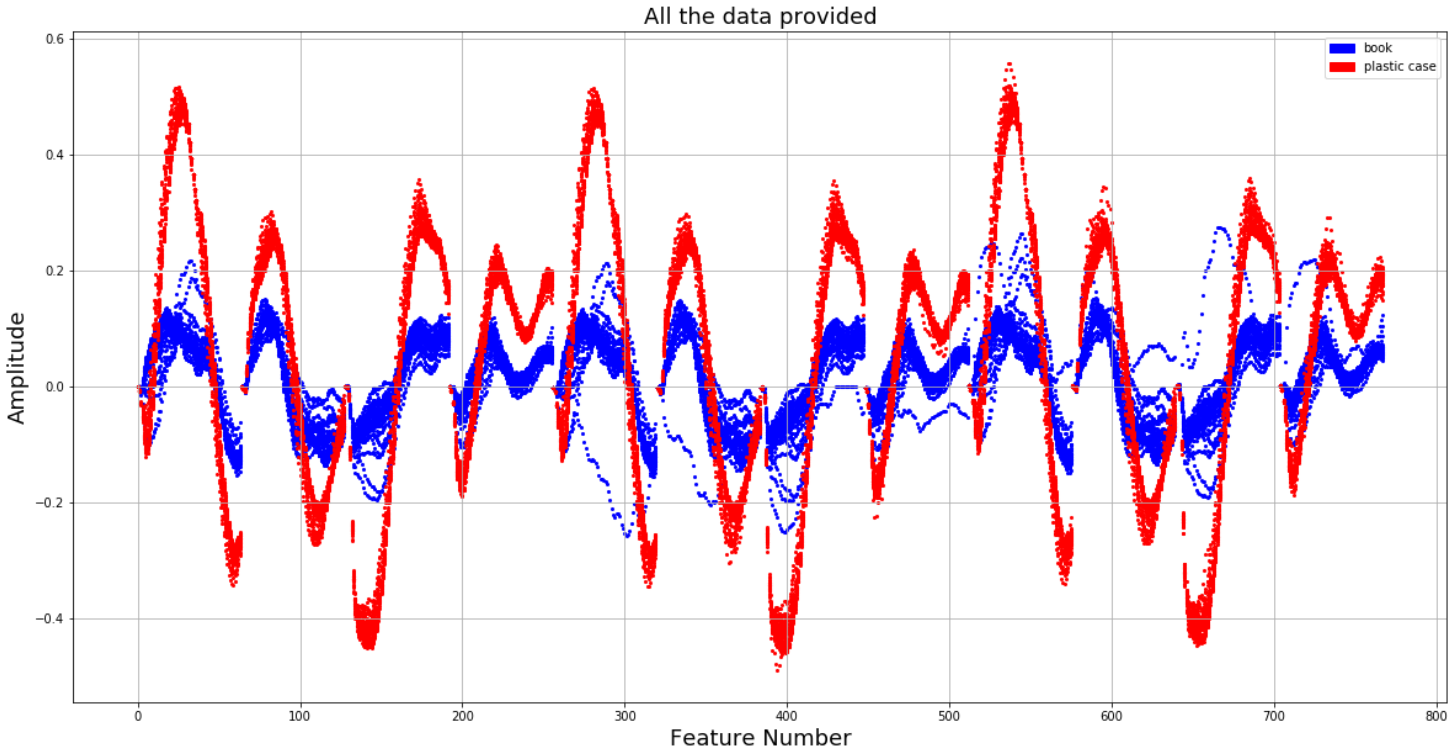

Figure 1. The figure above shows the data collected for all the features provided. It is color coded to show the difference in the classes.

The first 256 features represent the mean for each component in the four channels, followed by the 256 minimum values and the 256 maximum values recorded. From this initial observation is possible to see that overall the signal for the book class is generally weaker than the signal in the class plastic case across most of the features and in most samples.

It is worth noting that the small sample to feature ratio may mean that the data presented is not sufficient to train an accurate model given the high dimensionality of the data. Thus, further investigation into the correlation between the features and the output and also in between the features is required before feature selection can take place. This can be achieved by averaging all the samples across each feature and plotted the data on a channel by channel basis (the assumption made was that the first 64 features represented the mean features of the 64 components of channel 1). This result is shown below in Figure 2.

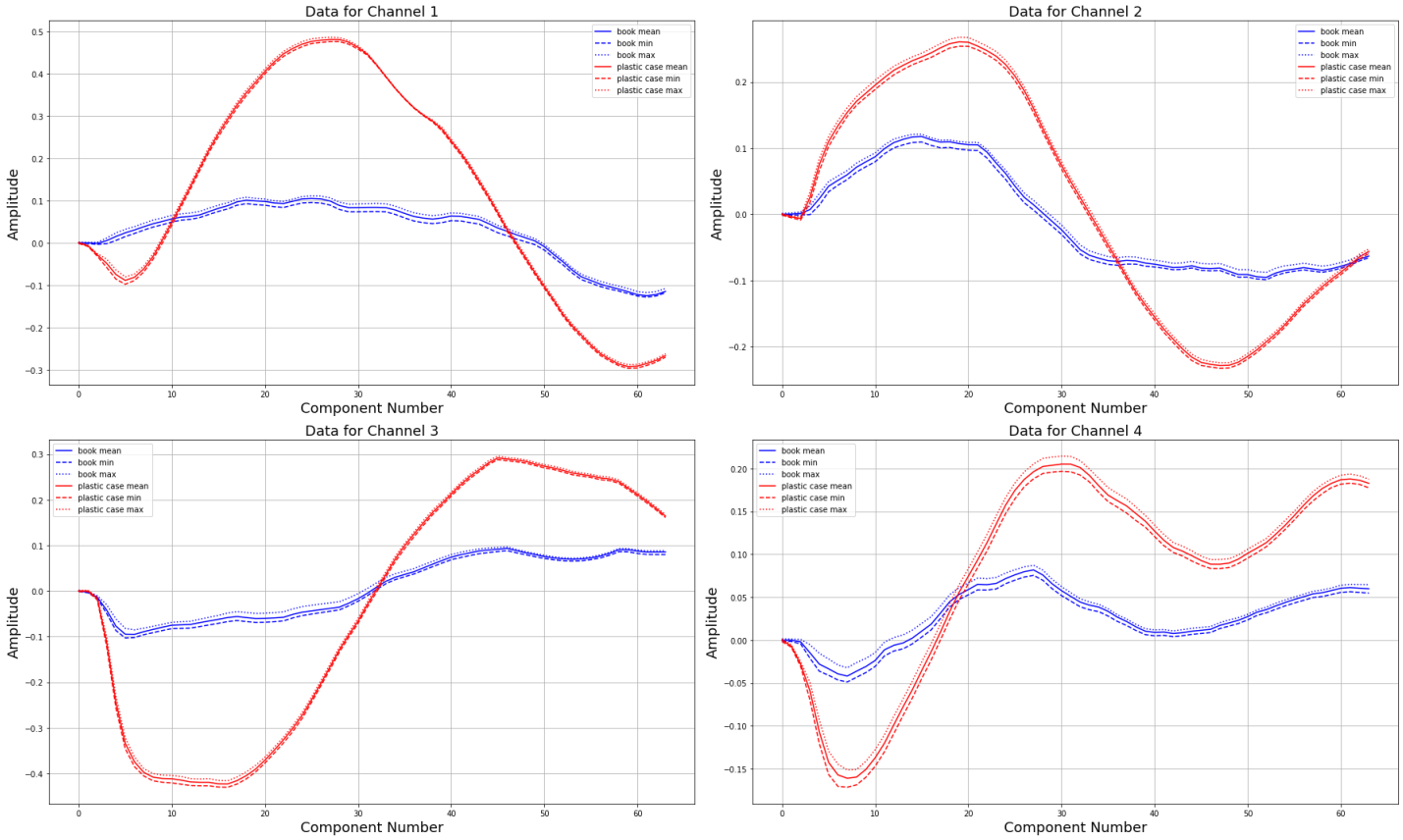

Figure 2. The figure above shows the averaged values for all the components split into a channel by channel basis. The filled line represents the average mean values, the dotted line represents the maximum values and the dashed line represents the minimum values. The color coding identifies each class. Blue is book and red is plastic case.

The line graphs shown above, display more clearly the difference in magnitude of the signals between both classes. As naturally expected, the maximum and minimum values plotted tend to follow the mean lines. These features are thus highly correlated with the mean and if the mean is highly correlated to the output then these features don't contribute to the precision of the model. These could be features that could be dropped without having a significant impact on the model's accuracy as they are highly correlated. In order to further observe the level of correlation of the mean values to the output label, a correlation curve using the Pearson Coefficient was plotted as seen below in Figure 3.

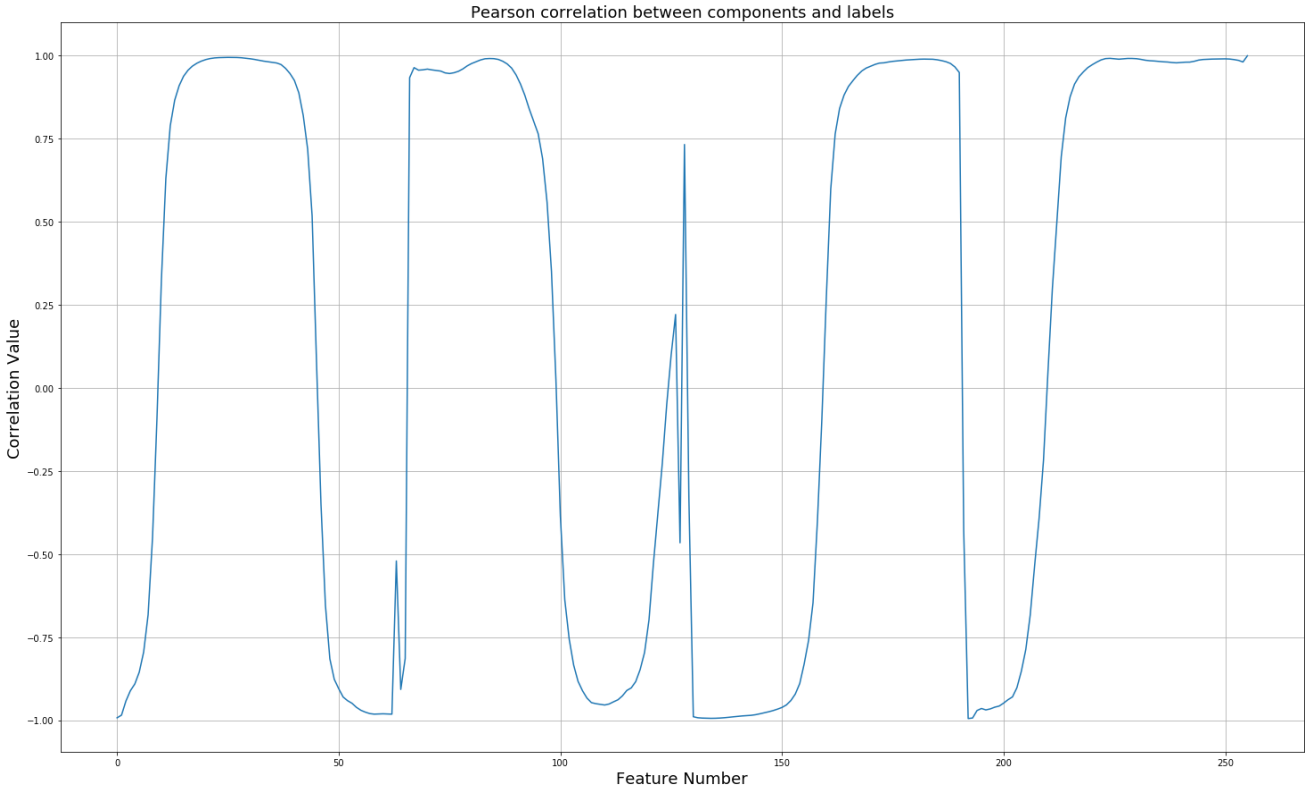

Figure 3. Figure showing the correlation of the mean features to the output. Correlation was calculated using the Pearson correlation coefficient. This takes the numeric encoding of the labels which explains the negative or positive correlations depending on which class as the highest strength in particular component.

The Pearson correlation coefficient measures the linear correlation between two variables. According to this plot, a strong correlation between the mean features and the classification is observed. In components where the signals can be easily discriminated, a strong positive or negative correlation coefficient is found depending on which class is greater in terms of the encoding value attributed to it. This indicates that some of these features could be dropped without sacrificing the accuracy of the model for this task. A correlation matrix was created to investigate the correlation between these mean features and it is shown below in Figure 4, focusing on channel 1.

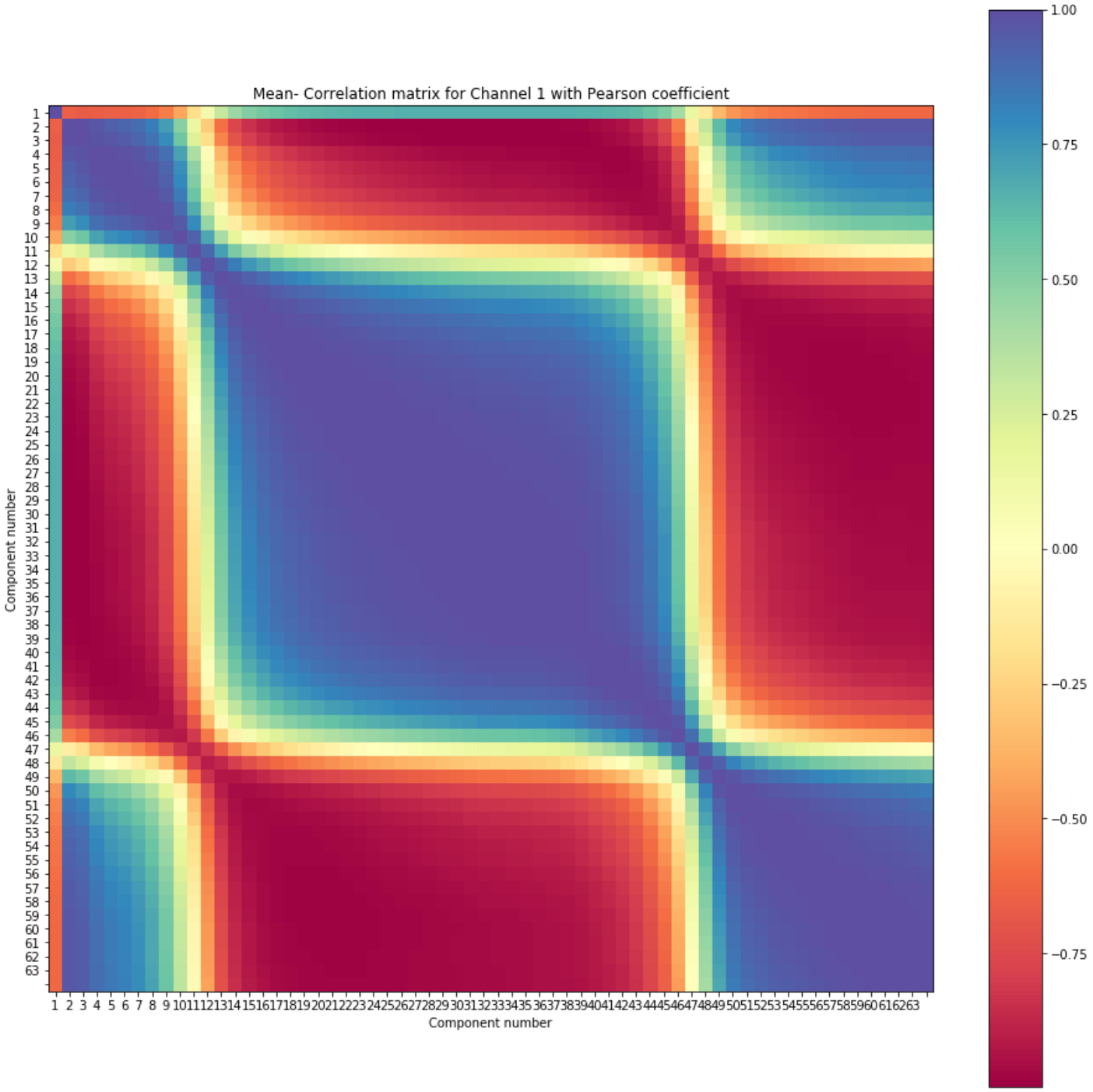

Figure 4. This illustrates the correlation between the mean components of channel 1. It uses the Pearson correlation coefficient as the correlation metric.

As expected, there is a high correlation between each component which follows the signal behavior closely (wave pattern). Thus, one of the conclusions reached is that a handful of features could be selected to train the model and still have a level of accuracy. This could potentially reduce the level of model complexity without sacrificing performance. Less inputs features would mean less computation required and potentially a faster prediction time.

In order to better visualize the data and get an understanding of how many dimensions the data could be reduced to, 2 component Principal Component Analysis (PCA) was conducted on the training data available. The results are displayed below in Figure 5.

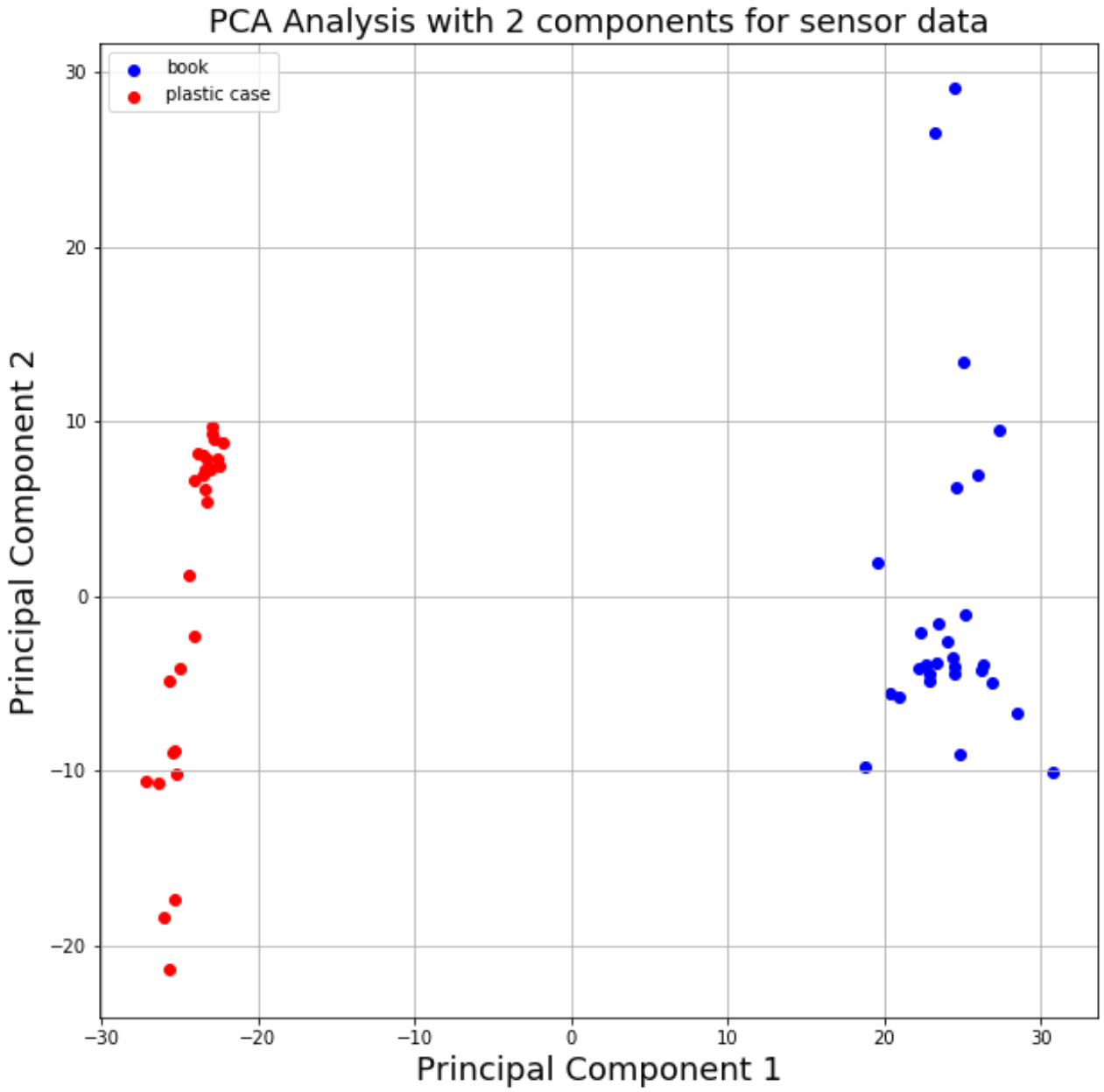

Figure 5. This figure shows the results of the Principal Component Analysis conducted using 2 principal components. The separation of the classes is very clear in this graph however it's worth noting the loss of information due to the dimensionality reduction process.

As clearly observed in Figure 5, the data is highly separable which means that a basic model such as Logistic Regression could be used to solve this classification task. These 2 principal components can account for about 88.2% of the variance in the data which further emphasizes the argument that feature selection would be effective. This clear separation also allows the training of a second model, a Support Vector Machine (SVM) based algorithm model as this is very useful when the data is relatively separable.

Preparing input and training classification models

Even though the 2 component PCA reduced dataset could be used to train the model, a different approach was used for feature selection. Only the mean values for the four channels were selected to train the model (the first 256 features). For the purposes of this task and given the small size of the dataset provided, a comparison between using both the means and the full dataset to train the models was conducted.

Feature scaling was implemented through Sklearn's StandardScaler to prevent the model from biasing towards features with high magnitudes. As mentioned before, two models were selected for training: Support Vector Machine and Logistic Regression. In order to further reduce the feature set, Lasso regularization was introduced to the Logistic model in order to punish less important features and further continue the process of feature selection. Given the low correlation rank of some of the features used, these techniques were highly encouraged. The standard *liblinear* solver was used since it performs well with small datasets.[1] The tuning hyper parameter for this model was C, since it controls the regularization strength of the model. Since a high regularization is preferred, low values for this attribute (0.01 and 0.1) as well as the default level (1).[2, p. 139]

Sklearn's C-Support Vector Classification (SVC) was used to implement the Support Vector Machine model. Given the clear separation of each class in the data, a simple linear kernel was implemented to keep model complexity low whilst allowing the use of this advanced technique. A range of values for the hyper parameter C were introduced since C controls the width of the margin for training the model. Since the data is clearly separable a high value of C could be used however, this would result in a less generalized model[2, p.147].

Both models were trained using K-Fold cross validation which divided the training data into 10 folds and tuned models to the hyper parameters specified. This was implemented using Sklearn's GridSearchCV feature. This was performed to evaluate the models during training without compromising the test data by selecting 10% of the data and creating a validation set. This validation set and the training data would change for each fold iteration. This feature tends to make better use of the data especially in this case where the dataset is small at the expense of making the training process more computationally expensive. The scoring metric chosen for selecting the model with the best parameters during this process was the Receiver Operating Characteristic – Area Under the Curve (RoC-AuC). This metric is a common tool to evaluate binary classifiers[2, p.91] The same process was applied to training sets (reduced and full).

Training and Testing results and Discussion

Both models performed well during training with both training datasets. Both algorithms registered a 1.0 RoC–AuC score as their best score in the 10-fold cross validation check. The results are shown below in Table 1 as well as the results of the test phase. There were however, differences in the training time where Logistic Classification was the fastest to train using the reduced training set (just means) in comparison to using the entire feature set. The SVM models on the other hand, registered a minor decrease in training time.

Model	CV Best Score (RoC-AUC)	Training Time	Testing Time	Overall Precision	Overall Recall	Overall F1-score
SVM	1.0	0.280	0	1.0	1.0	1.0
SVM – Means only	1.0	0.200	0	1.0	1.0	1.0
Logistic Classification	1.0	0.381	0	1.0	1.0	1.0
Logistic Classification – Means only	1.0	0.173	0	1.0	1.0	1.0

Table 1. The table above shows the results for the 2 models selected using different test sets. Note that given the way the training time was measured, it is not possible to compare between the two models. Additionally, the time will depend on the machine the Python script is running on.

The testing set was 30% of the overall data and the data was stratified during the split to ensure that the test data was balanced for both training and test sets. As expected, using only the means to train the models led to faster training and in this case, there was no decrease in accuracy. But in order to verify this observation, the model was evaluated using the "hidden" test data. Here, all models presented 100% accuracy with the testing time being too small to measure given the available tools. All models showed precision, recall and f1-score values of 1.0 for all the features.

Since the confusion matrix was the same for all the models used as they all had the same values, only a single example diagram was displayed to avoid redundant images. This is shown below in Figure 6:

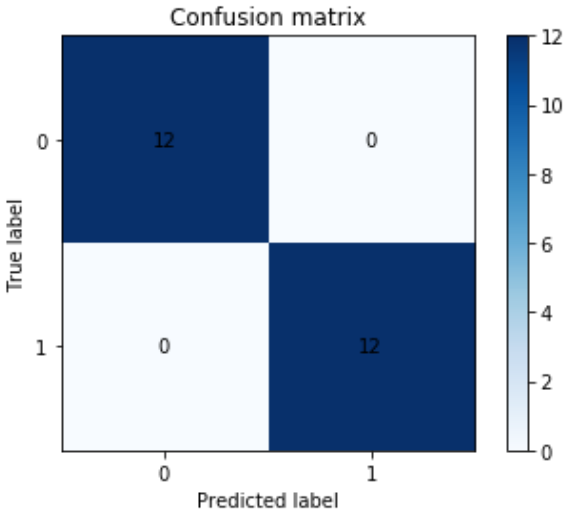

Figure 6. In this figure, Book is 0 and Plastic Case is 1. This confusion matrix demonstrates the performance of all the models (all 4 models trained showed the same results). A total of 24 samples were used for testing with an equal split,

This further reinforced the hypothesis that solely using the means features for this task would result in a model with faster training times and potentially faster prediction times as less important features would not be required to be processed and used. As previously observed, since the maximum and minimum values are highly correlated to the means values they don't add additional information.

SVM is very effective for this scenario where the number of dimensions is the greater than the number of samples, however care must be taken when selecting the kernel to avoid overfitting.[3] Thus it was decided that for predicting the unknown labelled data, the SVM approach trained with the mean features would be used. It is important to note that the model was retrained beforehand with the entire means set (both training and testing set) as it should increase the accuracy of the model.

It is important to note that choosing a Linear Support Vector Classification (LinearSVC) instead of the SVM could allow for more flexibility in this task. A linear model was applied to SVM which is already built in to the LinearSVC and thus this would enable the fine tuning of more parameters such as selecting the loss function (i.e. hinge). Whether this would result in a more accurate model is pure speculation.

Multiclass classification Task

Plotting and analyzing the data

A similar analysis process performed in the Binary Classification Task was implemented for the Multiclass classification task. Initially, the entire dataset was plotted as shown in the figure below:

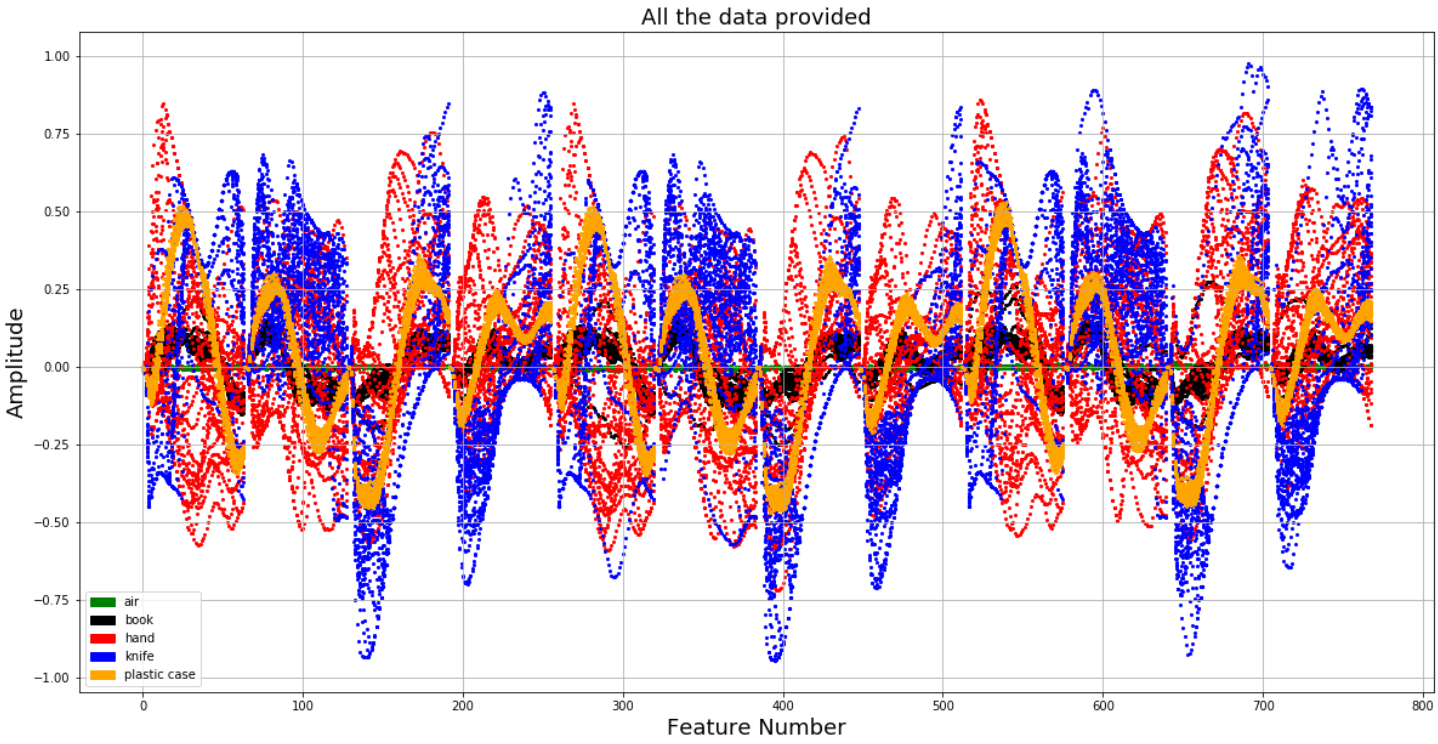

Figure 7. This figure shows all the data for all the features given in the training dataset. Each classification was color coded differently. The difference in magnitude can be seen for some classes across the components. The first 256 features represent the mean, followed by 256 minimum and 256 maximum of the components.

Although it is hard to visualize the data for each individual classification, certain conclusions can be drawn from the figure above. The Air class appears to have a small signal size throughout the entire feature set relative to the other components. Book and Plastic case classes appear to have unique magnitude ranges for some of the components. Additionally, the samples are less sparse in comparison to the samples for the hand and knife class which present a wide bandwidth and are consequently less discernible from the other three signals. However, a clearer visualization of each classification is required to further confirm these patterns hence, all of the feature sets were averaged for each classification and plotted on a channel by channel basis as seen in Figure 8 below:

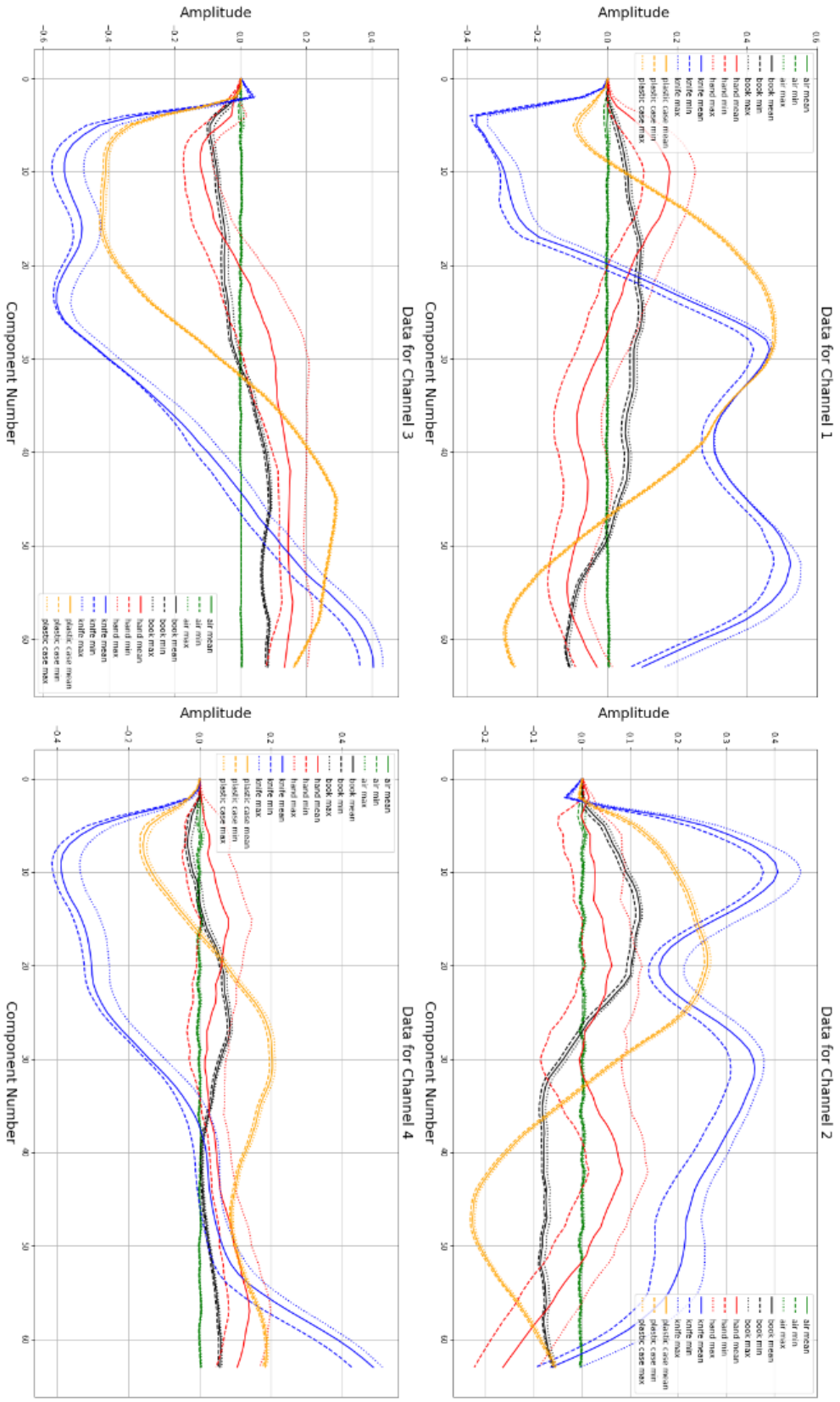

Figure 8. Average values of the mean, maximum and minimum features for each channel separated for each classification. Figure rotated to improve readability.

As seen in the graph above, the classes are highly discernible in certain components for all channels. The almost static nature of air is clearly observed and this is believed to be the baseline for the channel reading. Any deviations are very likely due to the noise in the sampling for this class. The maximum and minimum signals were also plotted for class and they appear to follow the mean values closely for book, plastic case and hand. However, these show different magnitudes for the hand and knife class and could potentially be used to help identify these classes. In certain components where the mean value is not sufficient to determine the class of the object, the maximum or minimum values would allow the algorithm to correctly identify certain classes as secondary metric. In Figure 8, this situation is observed in Channel 1 around the range of between component 25 and 35 where knife and plastic case are not clearly discernible when solely using the mean values. The difference between these two classes is in the minimum features for this range. Consequently, the previous strategy of only using the means features might not prove as effective as in the binary classification case. However, the argument can be made that either the maximum or minimum features can be selected since for the classes where they are relevant for discrimination (Hand and Knife) they tend to have different magnitudes. However, between these two features, the minimum features appear be generally more discernible from the mean for all classes. Thus, an interesting approach would be to just train the models with means and minimums features.

The correlation between components and the output labels was plotted in the figure below where the Pearson coefficient was calculated for the mean values.

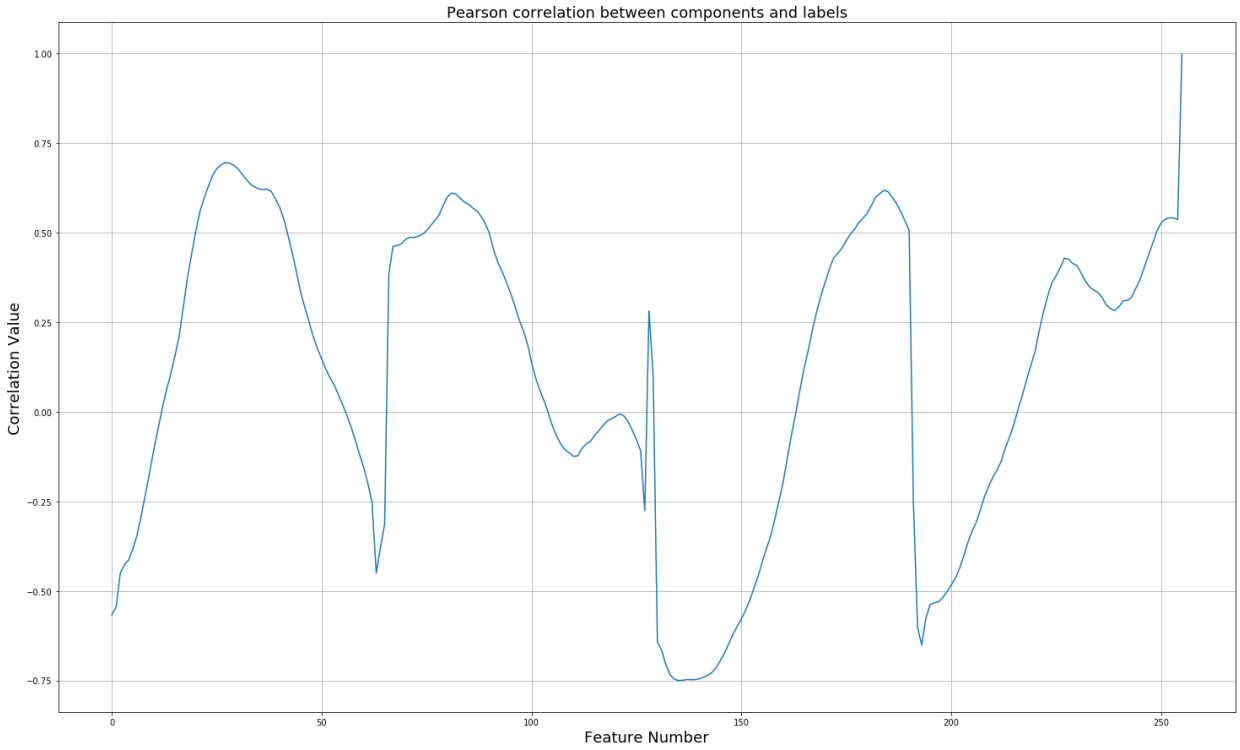

Figure 9. This figure displays the correlation of the means of the components with the output label. Compared to the Binary task, the components appear to be less clearly correlated, but correlated nonetheless which implies that machine learning can be conducted on the data.

Correlation with the output is observed in some components, however it is not as strong as in the binary case as the presence of more less separable classes skew the correlation value. This figure further reinforces the case for feature selection as the most correlated components could be used to train the model. A further investigation of the correlation between mean values in channel 1 was conducted to get a better understanding of how the classes behave throughout the components in that channel.

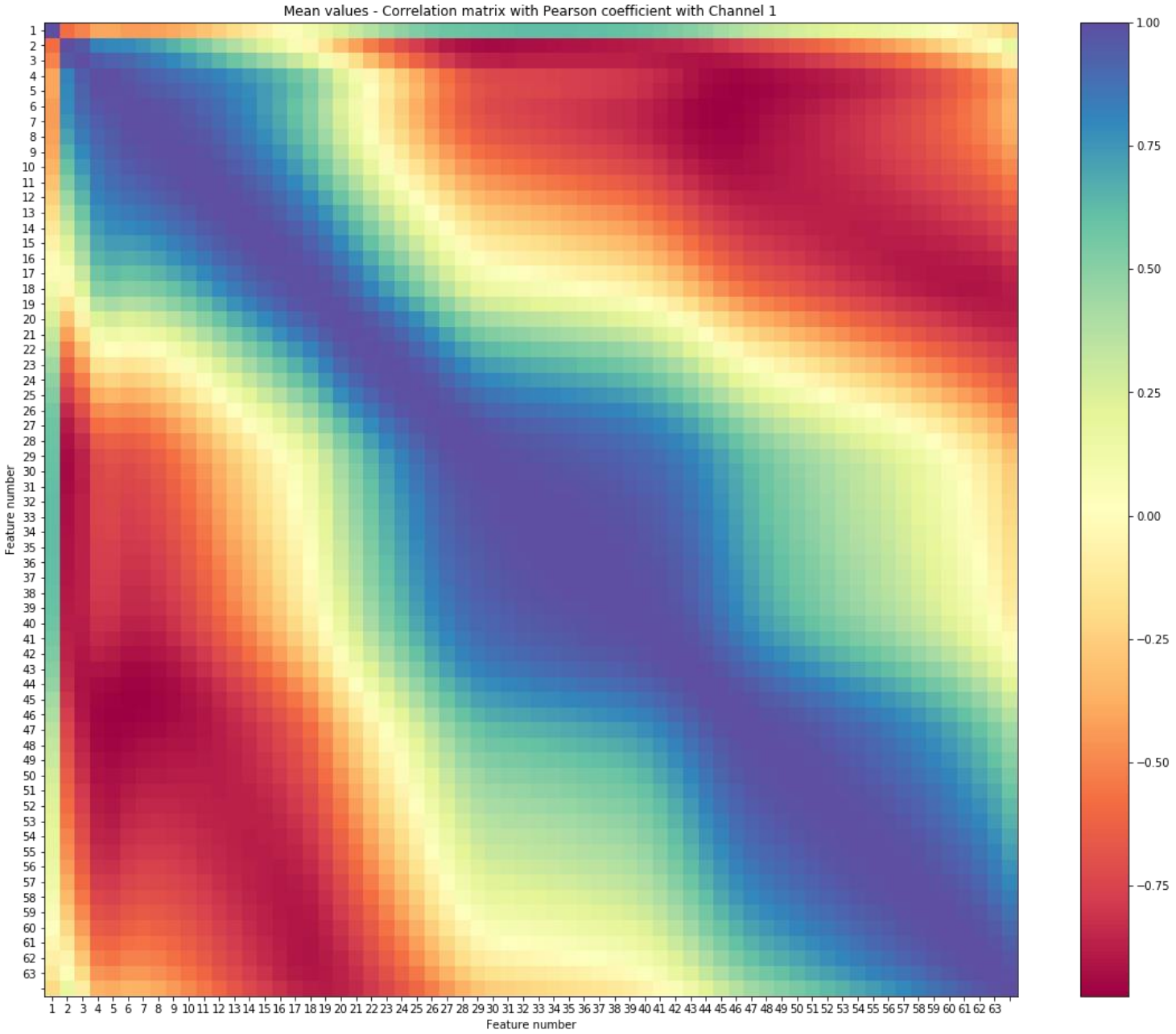

Figure 10. This figure displays the correlation of each component's mean feature for sensor 1. Notice the wave pattern.

As seen in the binary classification case, the means values appear to follow a wave pattern (gradual oscillation from positive to negative correlation or vice versa). This was expected given the shape of the signal for the classes throughout each component as seen in Figure 8.

In order to understand which features could be selected to train the model PCA analysis was conducted on the training dataset to check how much information would be lost if data was reduced to 2 principal components. This also helps to visualize the data in a 2-D manner. The results are shown below:

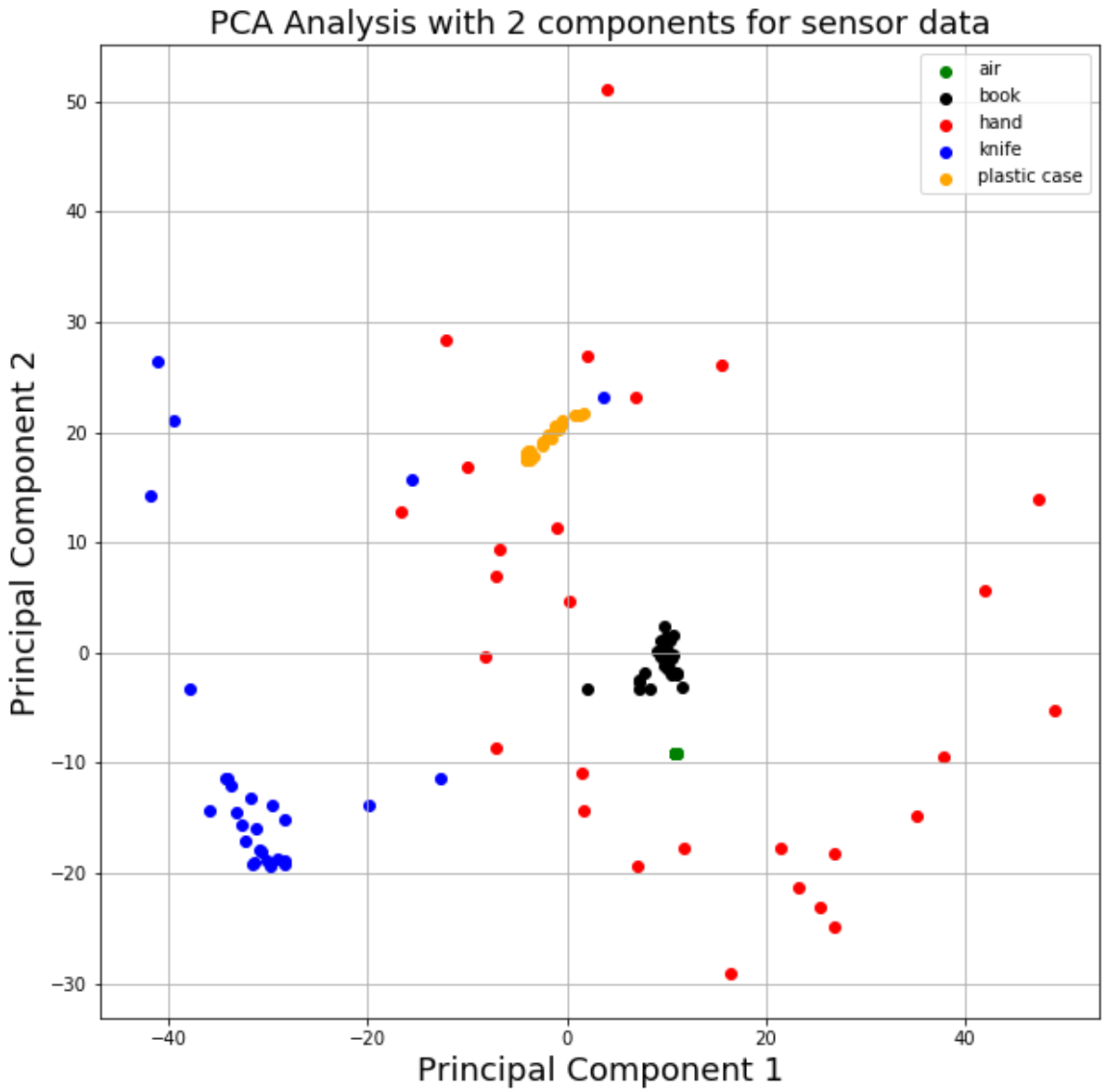

Figure 11. This figure displays the data after undergoing Principal Component Analysis. The high number of dimensions were reduced to 2 Principal components that accounted for 72.54% of the variance in the training data.

The observations made previously still hold with regards to how separable the classes air, book and plastic case are. The data for knife class appears to be clustered in a particular location whereas hand is more sparsely distributed. Both of these classes will be the most challenging for models to identify. These two principal components account for 72.54% of the variance in the information which means that using this reduced set would most likely lead to a very inaccurate model as too much information has been lost in the dimension reduction process. There are few outliers for both cases however, because there is still a

visible separation in the data. This leads to the conclusion that an SVM model should be attempted for this classification task as well as a more basic Logistic Softmax Regression model.

Preparing input and training classification models

As mentioned before the data was standardized using sklearn's StandardScaler library. Two sets were used for developing the models, one with the full training dataset and another dataset with just the means and minimums features for all the four channels.

Logistic Softmax Regression was used since it supports multiclass classification without having to train and combine multiple binary classifiers.[2, p. 140] This is a candidate model since the data is still separable with a few outlying cases. This required using a different solver to the standard *liblinaer*, using *lbfgs* instead as it is highly robust and it converges faster for high dimensional data, resulting in true multinomial logistic regression as opposed to using the default one-vs-rest option and creating multiple binary classifiers. This step increases the likelihood of that the Logistic Regression model's probability estimates will be better calibrated for this classification task.[4] However, this solver only support Ridge Regularization (L2) which does not reduce the number of important features to the scale of Lasso. [5] The same values for the regularization hyper parameter (C) used previously in the binary classification task were used for fine tuning the model (0.01, 0.1 and 1) to try to reduce the number of existing features and punish the less correlated ones.

An SVM based model called C-Support Vector Classification was used with a polynomial kernel as the linear option appeared unsuitable as the data did not appear linearly separable in some components. Nevertheless, the linear kernel was checked as the polynomial degrees used for fine tuning the model ranged from 1 to 3. Lower order polynomials were preferred to prevent the model from overfitting. A range of values for the hyper parameter C were introduced since C controls the width of the margin for training the model. Since the data is clearly separable a high value of C could be used however, this would result in a less generalized model[2, p. 147]. The same was applied to the hyper parameter gamma which controls the influence of a single training example.[3]

A 10-fold cross validation approach was used to train the model with the scoring metric set to f1-micro since this is the harmonic mean between precision and recall.[2, p. 86] Using F1 score helps to ensure both recall and precision scores are high. With regards to averaging method, micro-average was selected as it looks at individual true positives, false positives and false negative for the different sets. However, since there isn't a class imbalance in our test sample either micro or macro as an averaging is accepted for calculating the f1 score (multiclass). [6]

Training, testing and evaluating a model

Both models were trained with the two datasets and the results are shown below in Table 2.

Model	CV Best Score (F1-micro)	Training Time	Testing Time	Overall Precision	Overall Recall	Overall F1-Score
SVM	0.950	5.50	0.001	0.97	0.97	0.97
SVM – Reduced Set	0.950	4.64	0.009	0.95	0.95	0.95
Softmax Logistic Classification	0.964	6.874	0.000	0.95	0.95	0.95
Softmax Logistic Classification - Reduced Set	0.957	3.402	0.001	0.95	0.95	0.95

Table 2) Table showing the different model performance during the training phase and the results of the testing phase when test data is used.

As displayed in the table above, even though Softmax scoring the highest F1 score during cross validation training it achieved similar results in terms of overall precision and recall during the testing phase. In fact, SVM using the full dataset achieved the highest performance for testing even though it yielded a lower score during cross validation testing. The reduction in training time from using the reduced set was more significant in the Softmax Logistic Regression model. The best polynomial degree selected for both SVM models was 1 which meant that a linear kernel would be sufficient for this task. As expected, there was a minor decrease in overall precision in all the performance metrics when using the reduced dataset to train. However, in this particular case the time savings from the training step were not considered high enough to compensate for the performance loss. Displayed below are the confusion matrices for each model.

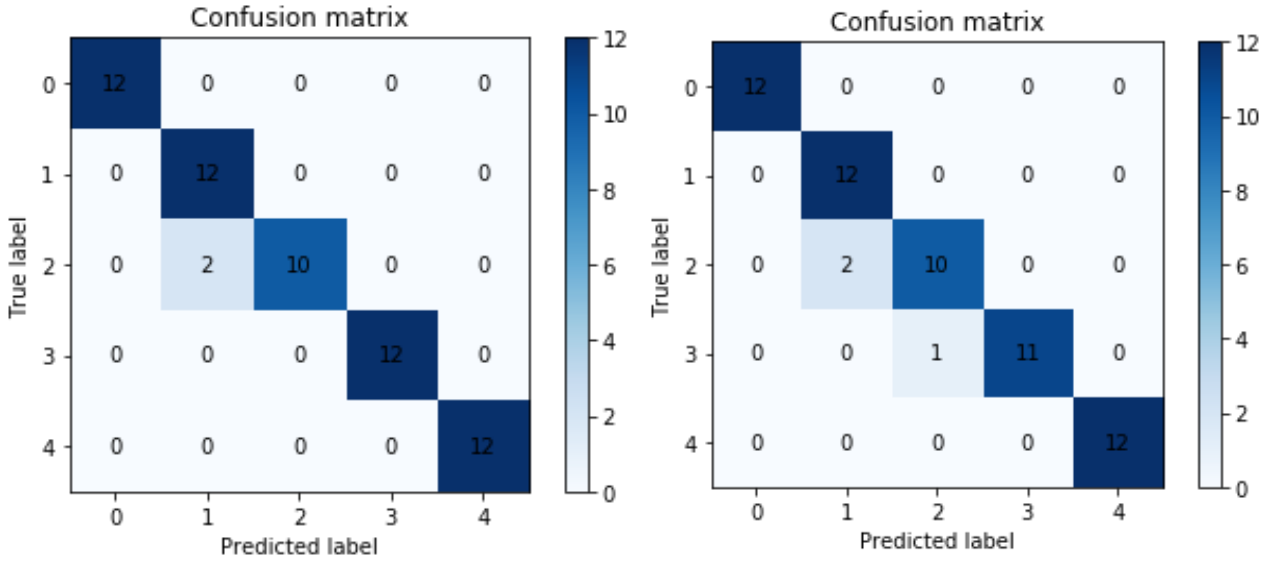

Figure 12. The figure displays the results of SVC model when using all the features (on the left) and the when selecting the means and minimums features for training (on the right). The models still perform well but using the entire feature set leads to a more precise model. Note that (0=Air, 1=Book, 2=Hand, 3=Knife, 4=Plastic case).

The confusion matrices above clearly demonstrate the accuracy drop from using the reduced dataset. It is worth noting that the classification label that is hardest to be accurate is the hand class as previously hypothesized during the plotting of the data. Most misclassifications were centered on the hand class and when the maximums features were removed, it became harder for the model to distinguishing these two classes. Thus, maximums and minimums features appear to help to model become precise.

The same pattern is observed here in the logistic regression model. The hand class is again the hardest to estimate however, the failed predictions tend to be focused on this class whereas in SVM, the knife class

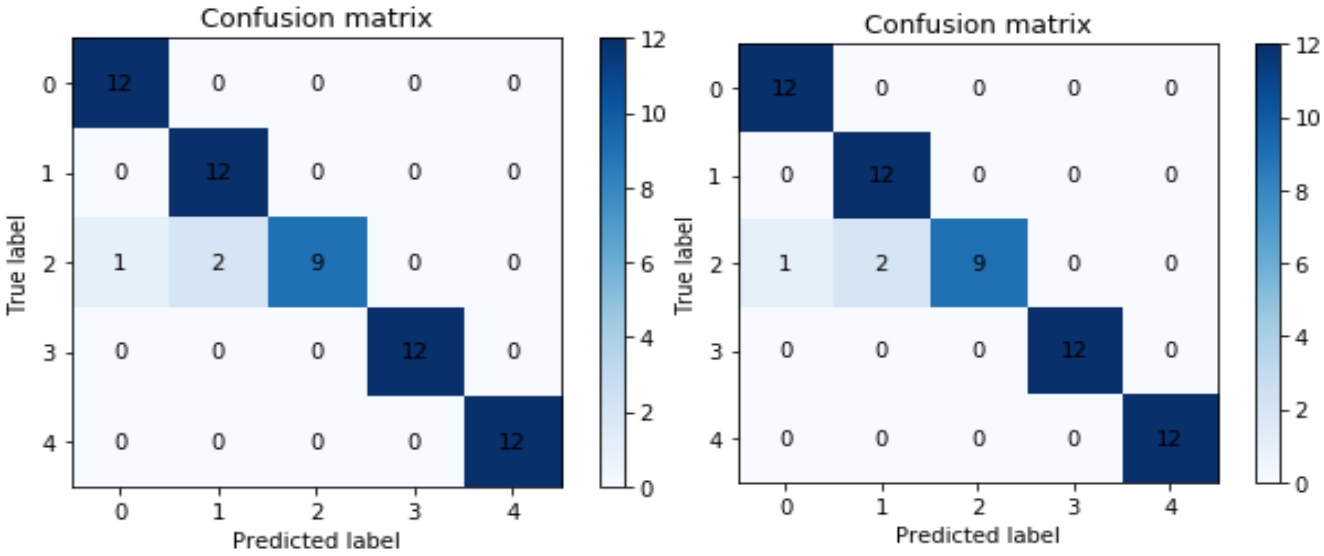

Figure 13. Figure showing the results for testing the Softmax Logistic Classification Model with both sets of data, entire feature set (on the left) and the reduced feature set (on the right hand side). There are no differences in the confusion matrices for these two models. Note that 0=Air, 1=Book, 2=Hand, 3=Knife, 4=Plastic case.

was misclassified once. There are also no changes in the accuracy of the Softmax Logistic Classification model when using the reduced training set. However, all the models appear to overall perform well with most of the classes registering 100% precision in the majority of the cases.

These results lead to the conclusion that the most effective model for this classification task is the SVM model trained with the entire feature set. This model was chosen to predict the unknown labelled data XToPredict after it was retrained with the full dataset (both training and test data).

Conclusion and Future work

In conclusion the SVM model was selected for both tasks as it performed well against logistic regression. In the binary case, it appeared to be the model of choice due to the wide separation however in the multiclass case, the efforts of implementing a kernel trick to prevent increasing complexity of the model were rendered pointless as the linear kernel was found to be the optimal parameter. Nevertheless, Logistic Regression was still suitable for the task however, it was less accurate in the multiclass case.

In terms of potentially future work it would be interesting to develop a third more advanced model using either a neural network or a random forest classifier to achieve a more precise result. Another interesting future work path would see the results of the model if train/test split was in the region of 50%. Additionally, it might be relevant to familiarize with Scikit's feature selection component as it can be a

powerful for deciding which features are of most interest. Nevertheless, the models presented in this report appear to have levels of precision and should be suitable for the task requested by this practical.

Feature selection should have occurred regardless of the nature of the data (max, min and mean) and a two-step process could have been followed for feature selection. Initially, all features that are not above a certain correlation threshold to the output would be removed followed by removing redundant features that show high levels of correlation between themselves. This could prove useful if the dataset was larger, however for the given dataset using the entire feature set appears to be the best approach.

References

[1] Scikit-learn, “Logistic Regression.” .

[2] A. Géron, *Hands-On Machine Learning with Scikit-Learn*, 1st ed. 2017.

[3] Scikit-learn, “Support Vector Machines.” [Online]. Available: https://scikit-learn.org/stable/modules/svm.html#svm . [Accessed: 14-Apr-2019].

[4] Scikit-learn, “Linear Model.” [Online]. Available: https://scikit-learn.org/stable/modules/linear_model.html. [Accessed: 14-Apr-2019].

[5] N. Anuja, “L1 and L2 Regularization Methods,” 2017. [Online]. Available: https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c. [Accessed: 14-Apr-2019].

[6] A. Swalin, “Choosing the Right Metric for Evaluating Machine Learning Models — Part 2,” *Medium*, 2018. [Online]. Available: https://medium.com/usf-msds/choosing-the-right-metric-for-evaluating-machine-learning-models-part-2-86d5649a5428.

Appendix

Multiclass Classification for unlabeled data provided	
Sample number	Predictions
1	2.0
2	2.0
3	2.0
4	2.0
5	2.0
6	2.0
7	2.0
8	2.0
9	2.0
10	2.0
11	0.0
12	0.0
13	0.0
14	0.0
15	0.0
16	0.0
17	0.0
18	0.0
19	0.0
20	0.0
21	3.0
22	3.0
23	3.0
24	3.0
25	3.0
26	3.0
27	3.0
28	3.0
29	3.0
30	3.0
31	1.0
32	1.0
33	1.0
34	1.0
35	1.0
36	1.0
37	1.0
38	1.0
39	1.0
40	1.0
41	4.0
42	4.0

43	4.0
44	4.0
45	4.0
46	4.0
47	4.0
48	4.0
49	4.0
50	4.0

Binary Classification for unlabeled data provided	
Sample number	Predictions
1	1.0
2	1.0
3	1.0
4	1.0
5	1.0
6	1.0
7	1.0
8	1.0
9	1.0
10	1.0
11	0.0
12	0.0
13	0.0
14	0.0
15	0.0
16	0.0
17	0.0
18	0.0
19	0.0
20	0.0