CIS4910 Final Report

Nathaniel Johnson - 1086286

Contents

Introduction

This research project had two objectives. The first was to compare the classification ability of several machine learning models. The models involved in the comparison were:

1. The Pattern Discovery model[1]
2. Scikit-learn's decision tree[2]
3. Scikit-learn's decision tree after applying an adjusted residual pruning strategy
4. Support Vector Machine[3]

We were particularly interested in determining if the adjusted Pruned Decision Tree (PDT) would retain the training and classification speeds of the Unpruned Decision Tree (UDT) and the higher accuracy and F1 score of the Pattern Discovery (PD) model and Support Vector Machine (SVM) classifier. The models were trained on a large, labeled set of ergonomic data, and had their classification accuracies and F1 scores compared. While comparing the models, we also worked on a second objective; to discover interesting patterns in the ergonomic data.

For this project, the ergonomic data was collected and provided by Dr. Nancy Black of the University of Moncton. The data was collected during an investigation into the ergonomics of a new sit/stand/cycle desk prototype. The investigation involved 25 participants who were connected to a CAPTIV sensor while using the sit/stand/cycle desk during their normal work day. The data consists of the readings from 25 different sensors within the CAPTIV device over the course of around 10 minutes, with 32 readings every second. Each reading is composed of the measurements taken by the sensors at that moment in time, each of which measures a specific angle of the neck, shoulders, back or hips. The features for our data come from the 18 of these sensors, representing the angles of the head, neck, shoulders, and back. The labels for the data come from a survey each participant completed, in which they gave a numerical value to the pain or stiffness they felt in different parts of their bodies after the end of the work day.

Methods

Tools

All scripts were written using Python 3.11.5, NumPy v1.24.3, Pandas v2.1.1, Scikit-learn v1.3.0, and Matplotlib v3.8.0. Data preprocessing, model training, and evaluation (excluding PD) were done on WSL2's Ubuntu 20.04 distribution. PD was trained and evaluated on SHARCNET's GRAHAM cluster.

Preprocessing

Preprocessing began by filtering out all participants who did not have DBG CAPTIVE data, or did not have an associated score for neck pain for the DBG CAPTIV experiment. Of the 25 participants, 9 did not meet the criteria and were removed (01, 02, 05, 08, 11, 15, 18, 21, 22, 25), leaving 16 for the analysis. The neck pain scores on a scale from 1-10 from the remaining participants were bucketed into 3 labels: Low [1-3], Medium [4-7], and High [8-10]. This was done to improve generalization by aiding to remove some of the variance in the subjectivity of the pain ratings. Of the 16 participants, 12 were labeled "Low", 3 were labeled "Medium", and only 1 was labeled "High".

The raw participant DBG CAPTIV data from the remaining participants were transformed into "windows", which represent a movement of the head, shoulders, or back over the span of a second. For a specific participant, the script creates windows by iterating through a column and selecting every other value until 16 values are selected. It then transforms these values into a row, indexed by the start time of the first selected observation, and suffixed with the appropriate label value for that participant's neck pain score. Each column denotes the offset in seconds for that reading from the start time. The script then skips half a second in time from the previous windows start time, before repeating the process. By skipping half a second, the windows overlap. This ensures that no movements are missed, but also that individual readings are not oversampled. This process is repeated until no more one second long windows can be created. The windows from that particular column for that participant are then saved to a .csv file. This process is repeated for each column, and for each participant.

After the windows have been created, another script cleans them. The sensors in the CAPTIV device did not always perform as expected and would occasionally fail to transmit their readings. This resulted in patches of zeros in the data, some large and some small, depending on how long the sensor outage lasted. Generally, this was no more than a few seconds. Cleaning the data in this sense refers to the handling of these zero values. For each set of windowed data, the script would remove any windows which represented second long intervals of failed transmissions. Then, any remaining zeros had their values replaced using linear interpolation as a way of filling in the gaps.

After the data had been cleaned, another script folded the data into training and testing sets using 4-fold cross-validation. This was done on a per feature basis, resulting in 4 folds per feature. For a given fold, the script partitioned 4 of the 16 participant's windows to be used for testing, and the remaining 12 participant's windows to be used for training. This is repeated 4 times, in such a way that no participant is used for testing more than once. By using the full data of each participant in each fold, we kept our experiment between participants, rather than within participants. This allowed us to minimize any carry-over effect the temporal data may have. Below is a table describing which testing fold each participant belongs to, and consequently, which training fold they are not a part of.

Fold Number	Fold 1	Fold 2	Fold 3	Fold 4
Participants	03 08 09 16	04 10 12 17	06 19 13 23	07 20 14 24

Table 1. Participants by training fold.

After an initial attempt at fitting the models resulted in classifiers with very high accuracy but low F1 scores, we realized that the imbalance in the labels in the training data was causing the models to classify nearly every window as 'Low'. This resulted in nearly all of the "Low" windows in the testing fold being correctly classified, but very few of the "Medium" and "High" windows being correctly classified. To address this issue, we used random undersampling to rebalance the training folds for each feature, reducing each model's bias towards the "Low" label.

Classifiers and Evaluation

The decision tree classifiers were created using the Decision Tree class from the scikit-learn python library, and were limited to a maximum depth of 15. This was done to improve training times and reduce overfitting. Adjusted residual pruning was applied using a custom-built library. The SVM was created using the SVC class from scikit-learn, configured to use a polynomial kernel with a degree of 3, and was run for 10,000 iterations. The PD model was configured to only search the event space for positive events and to use a maximum order of 4. Attempts were made to run training with a maximum order of 9, then 5, but runtimes were too long and resulted in the program being terminated before results could be achieved.

Training was performed similarly for each model. For each fold of each feature, the models were fit using a training fold, then evaluated using the associated testing fold. The accuracy and F1 score were recorded for each fold and were used to find the mean accuracy and F1 score per feature, by classifier. Truncated tables for the Accuracy, F1 score, and associated variances for each feature are provided with each classifier in the results section. Full tables are provided in the appendix. The fold metrics were also used to find the mean accuracy and F1 score for each group of participants making up that testing fold, across all features for the UDT, PDT, and SVM classifiers.

Results

Unpruned Decision Tree

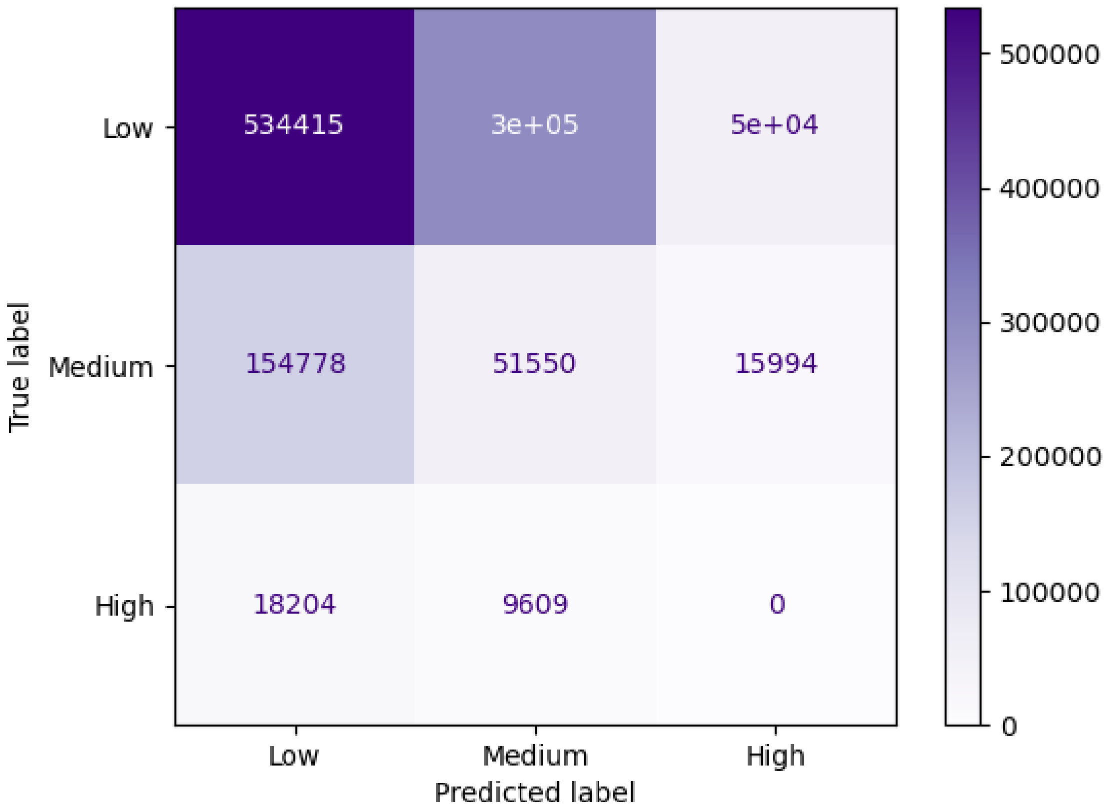

Fig. 1. Combined confusion matrix for all features - UDT

Feature	Mean Accuracy (Variance)	Mean F1 (Variance)
Angle Shoulder (Right) (Projection) Flex/Ext	0.595 (0.016)	0.323 (0.006)
Angle Shoulder (Left) Vertical Rotation	0.568 (0.003)	0.300 (0.003)
Angle Neck Flex/Ext	0.497 (0.000)	0.287 (0.003)
Angle Shoulder (Left) (Projection) Flex/Ext	0.520 (0.002)	0.284 (0.003)
Angle Shoulder (Right) Horizontal Rotation	0.505 (0.007)	0.279 (0.002)

Table 2. Top 5 features with the greatest F1 score - Unpruned Decision Tree

The UDT reported the greatest accuracy for Angle Back Rotation (accuracy=0.69, F1=0.273) and the greatest F1 for Angle Shoulder (Right) Projection Flexion Extension (accuracy=0.595, F1=0.323). Across all features, the UDT had a mean accuracy of 0.520 and a mean F1 score of 0.266.

Support Vector Machine

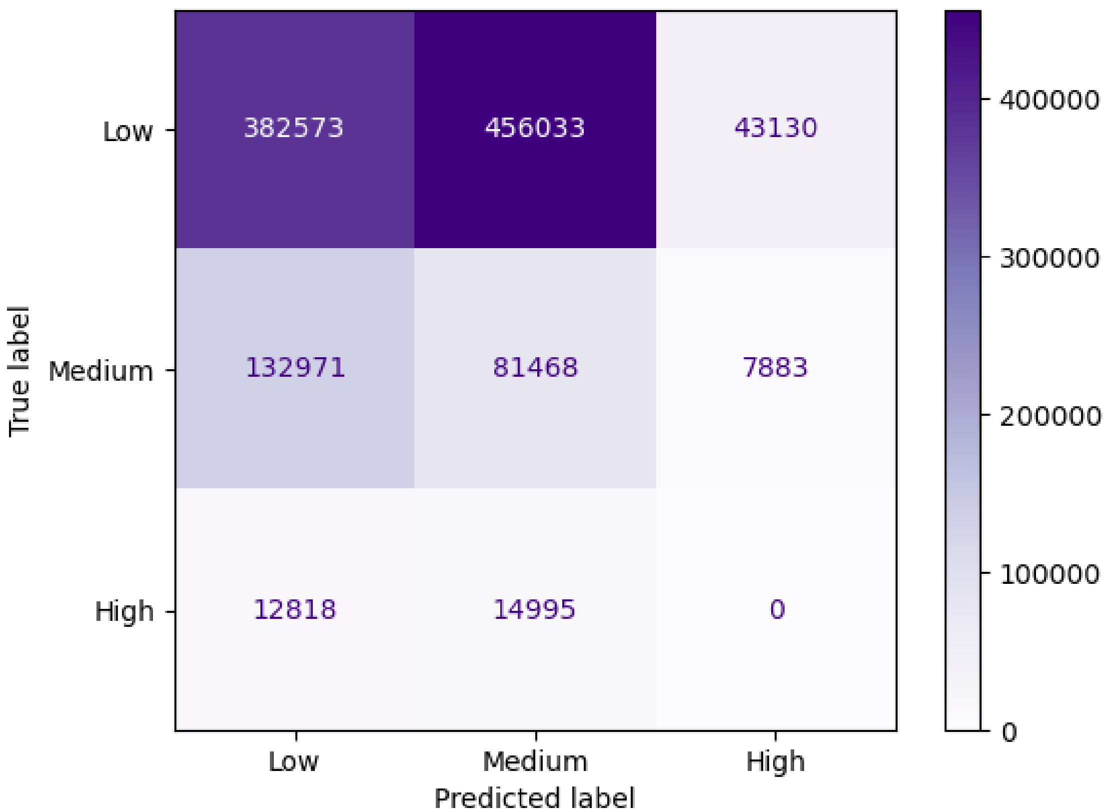

Fig. 2. Combined confusion matrix for all features - SVM

Feature	Mean Accuracy (Variance)	Mean F1 (Variance)
Angle Shoulder (Left) Horizontal Rotation	0.516 (0.058)	0.331 (0.034)
Angle Back Rotation	0.755 (0.153)	0.327 (0.029)
Angle Shoulder (Left) (Projection) Abd/Add	0.728 (0.063)	0.277 (0.003)
Angle Shoulder (Left) Vertical Rotation	0.569 (0.075)	0.270 (0.004)
Angle Neck Lateral Flexion	0.614 (0.109)	0.253 (0.005)

Table 3. Top 5 features with the greatest F1 score - SVM

The SVM reported the highest accuracy for Angle Back Rotation (accuracy=0.755, F1=0.327), and highest F1 score for Angle Shoulder (Left) Horizontal Rotation (accuracy=0.516, F1=0.331). Across all features, the SVM had a mean accuracy of 0.412 and a mean F1 score of 0.207.

Pruned Decision Tree

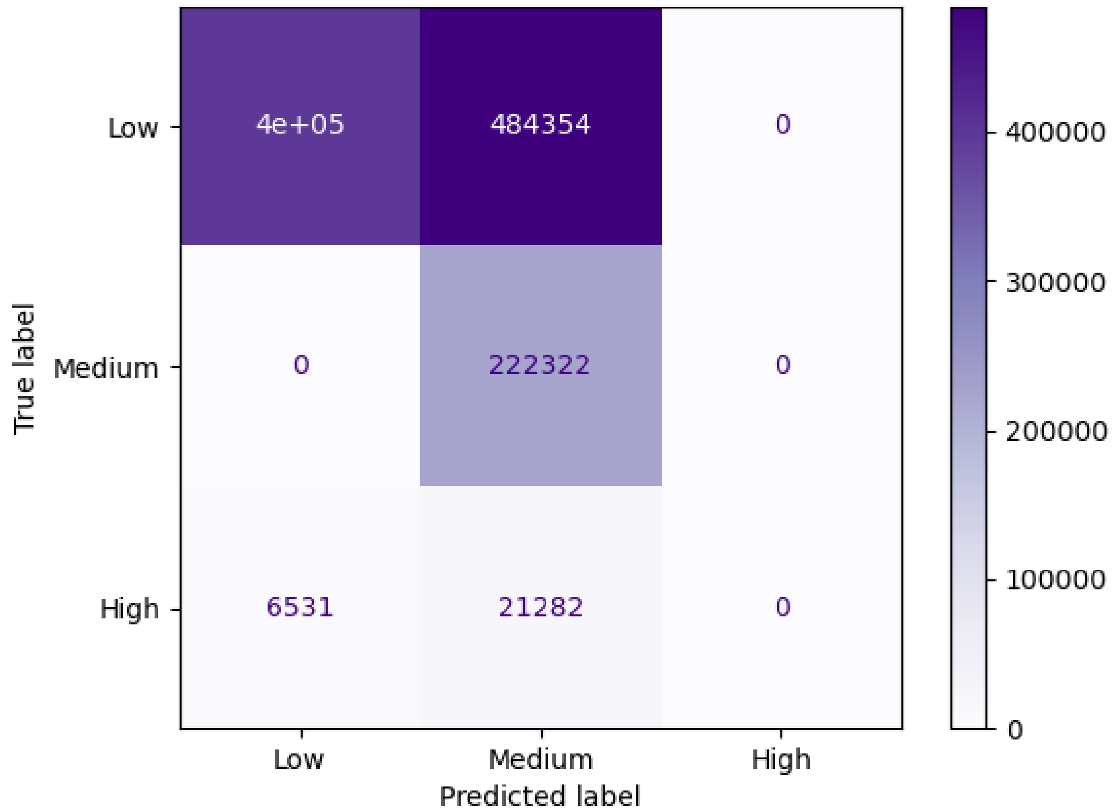

Fig. 3. Combined confusion matrix for all features - PDT

Feature	Mean Accuracy (Variance)	Mean F1 (Variance)
Angle Shoulder (Right) (Projection) Abd/Add	0.705 (0.118)	0.623 (0.192)
Angle Neck Rotation	0.678 (0.181)	0.610 (0.219)
Angle Neck Lateral Flexion	0.665 (0.177)	0.606 (0.219)
Angle Shoulder (Right) Vertical Rotation	0.662 (0.189)	0.603 (0.225)
Angle Shoulder (Left) (Projection) Abd/Add	0.678 (0.097)	0.513 (0.112)

Table 4. Top 5 features with the greatest F1 score (excluding Angle Back Rotation) - PDT

The PDT reported a perfect accuracy and F1 score for Angle Back Rotation. Upon further inspection, every training fold for the Angle Back Rotation feature contained exclusively observations labeled Low. The PDT also boasted the best average accuracy across all features at 0.543, and best average F1 score across all features at 0.457. Removing the outlying Angle Back Rotation feature scores brings the mean accuracy down to 0.516 and F1 to 0.425.

Pattern Discovery

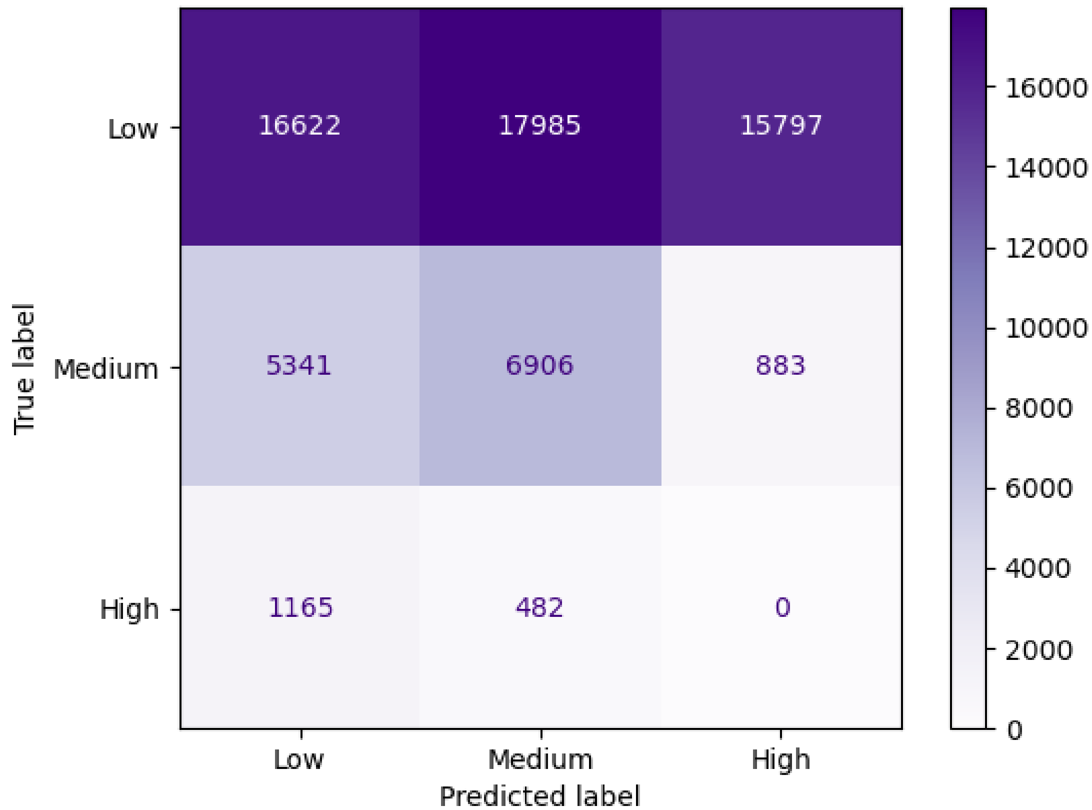

Fig. 4. Confusion Matrix for Angle Neck Flex/Ext - PD

Feature	Mean Accuracy (Variance)	Mean F1 (Variance)
Angle Neck Flex/Ext*	0.384 (0.022)	0.257 (0.015)
Angle Back Forward Flexion*	0.521 (0.085)	0.229 (0.011)
Angle Back Lateral Flexion*	0.380 (0.001)	0.218 (0.001)
Angle Shoulder (Left) (Projection) Abd/Add	0.428 (0.031)	0.215 (0.002)
Angle Back Rotation*	0.488 (0.111)	0.197 (0.018)

Table 5. Top 5 features with the greatest F1 score - PD
* - value was reported using unbalanced training data

The PD model was only able to be trained on a subset of the available features, as its expensive runtime meant training lasted several hours, even with configuring it to run with a relatively low order and to onlysearch the event space for positive events. PD was evaluated on the following features; Angle Neck Flex/Ext, Angle Back Forward Flexion, Angle Back Lateral Flexion, Angle Shoulder (Left) Projection Abd/Add, Angle Back Rotation, Angle Shoulder (Left) Horizontal Rotation, and Angle Neck Rotation.

Furthermore, Angle Back Forward Flexion, Angle Back Lateral Flexion, Angle Back Rotation, and Angle Neck Flex/Ext were all evaluated before the data had been balanced**.** PD reported the highest accuracy for Angle Back Forward Flexion, and the highest F1 score for Angle Neck Flex/Ext. Across all features, PD had an average accuracy of 0.399 and an average F1 of 0.208.

The features which were frequently among the five highest F1 scores were Angle Back Rotation (3 times), Angle Shoulder (Projection) Abd/Add (3 times), Angle Shoulder Vertical Rotation (3 times), Angle Neck Flex/Ext (2 times), Angle Neck Lateral Flexion (2 times), Angle Shoulder Horizontal Rotation (2 times).

Among the groups of participants, fold accuracy and F1 scores were very similar. The SVM classifier performed best when using the data in the first fold. The UDT and PDT both performed best when using the 4th fold. The differences in performance between folds were not significantly different for accuracy (ANOVA, F=0.49 p=0.70) or for F1 (ANOVA, F=0.12, p=0.95).

Classifier	Fold 1 Accuracy / F1	Fold 2 Accuracy / F1	Fold 3 Accuracy / F1	Fold 4 Accuracy / F1
SVM	0.517 / 0.253	0.312 / 0.167	0.376 / 0.180	0.422 / 0.219
UDT	0.502 / 0.277	0.514 / 0.263	0.523 / 0.266	0.533 / 0.260
PDT	0.489 / 0.366	0.509 / 0.436	0.520 / 0.432	0.625 / 0.563
Mean Total	0.503 / 0.299	0.445 / 0.289	0.473 / 0.293	0.527 / 0.347

Table 6. Mean Accuracy & F1 score by fold, per classifier.

Using the mean F1 scores from each fold, SVM, UDT and PDT were compared using Tukey's HSD following an ANOVA to determine if the differences between the mean F1 scores of the classifiers were significantly different (ANOVA, F=23.21, p<0.005). Tukey's HSD reported that the differences in F1 score between SVM and UDT were not significantly different, but that the differences in F1 score between the PDT and SVM, as well as the difference in F1 score between the PDT and UDT were significantly different (p<0.01 for both).

Interpretation

Given the low F1 scores from each of the classifiers, we can conclude that our models were not successful in determining a clear association between neck pain and angles of the head, neck, shoulders, and back. While SVM had around 70-75% accuracy for some features, the associated F1 scores were only around 0.25-0.35, indicating that the classifier accuracy was likely inflated due to the unbalanced distribution of labels in the data. The PDT provided the greatest F1 scores, but even its highest score was only just over 0.62. The models predominantly classified windows as either being ‘Low’ or ‘Medium’, and, except for PD, rarely attempted to classify windows as ‘High’. Interestingly, not a single classifier was able to correctly classify a ‘High’ window. This may simply be an issue with the distribution of the labels, as only a single patient gave a ‘High’ score for neck pain. For each of the models, the majority of false classifications were a result of classifying a ‘Low’ window as ‘Medium’. Looking at individual classifiers, the UDT and SVM frequently mislabel windows as being ‘Low’ when they actually were ‘Medium’, whereas the PDT never made that mistake. On the other hand, the PDT never attempted to classify a window as ‘High’. This suggests that the pruning algorithm was able to remove branches which would have resulted in false ‘Low’ classifications for ‘Medium’ windows, but also removed any branches which would have resulted in ‘High’ classifications. Seeing as the BDT failed to accurately classify any of the ‘High’ windows, this worked to the advantage of the PDT. PD was more likely to classify windows as being ‘High’ than other classifiers. However, PD seemed to randomly guess at the true label of ‘Low’ windows, and never correctly identified a ‘High’ window. It is possible that PD’s performance was impaired by the low maximum order it was configured with. With more time, it would be interesting to explore higher order configurations of PD on this data.

Overall, the PDT performed the best, reporting the highest average F1 scores, and some of the highest accuracies. This is encouraging, as it lends strength to the hypothesis that the adjusted residual based pruning strategy can improve the classifiers performance when trained on noisy data. In terms of training speeds, the PDT was the second fastest, only slightly slower than the UDT, and slightly faster than the SVM.

What I’ve Learned

Became proficient at analyzing and interpreting data and results.

The second half of this project has given me the opportunity to develop my data analysis skills. When building the experiment, I had to determine which metrics and statistical methods I was going to use to compare the models, as well as to determine if the models had successfully found patterns in the data. Applying those statistical methods and attempting to interpret the results pushed me to try and understand my data and come up with new hypotheses, which in turn allowed me to dive deeper into the problem. I also gained experience using pandas and matplotlib to create graphs and tables to aid with analysis.

The analysis section for this project was done late into the semester, and while I think I certainly made progress towards this goal, I’m not sure if I got the amount of experience I was after. Had there been time to conduct an even deeper investigation, such as looking at model performance when trained using multiple features, I feel that I would have fully achieved this goal. Unfortunately with the amount of work from other courses, this was not feasible, and so I feel this goal was really only partially achieved.

Gained experience cleaning and transforming data.

Over the course of the project, I've gotten a lot of hands-on experience with Pandas, an open-source data manipulation and analysis tool. Before the start of the project, I still felt very uncomfortable and slow with Pandas. Since starting this project, I've used Pandas for all of my data processing, and I'm happy to say that I feel I have become very efficient with pandas. This has significantly increased the rate at which I can accomplish data manipulation tasks, and has made the normally tedious task of data preparation rewarding. I was able to create scripts to perform sampling, run the experiment, and create tables much, much faster than I would have been able to at the start of this project. Seeing how far I have come with this skill makes me certain I have achieved this goal.

Practiced technical communication skills and knowledge mobilization.

With the reports I have written this semester, as well as the weekly discussions with Andrew, I have had the opportunity to practice translating new and complex topics into understandable contexts. I've had the opportunity to learn new terms and have previous misunderstandings corrected. This final report has been the most difficult to write, but also the most rewarding, as it's challenged me to think about how to present my procedure and my results in a way that is repeatable and understandable. While I have not been given feedback on my writing so far, nor have I had to do a presentation, I feel that with what I have written, along with the feedback I will receive, there is enough accomplished for me to say that I have achieved this goal.

Appendix

Feature	Accuracy	F1	Acc Var	F1 Var
Angle Shoulder (Right) (Projection) Flex/Ext	0.596	0.324	0.016	0.006
Angle Shoulder (Left) Vertical Rotation	0.569	0.301	0.003	0.003
Angle Neck Flex/Ext	0.497	0.287	0	0.003
Angle Shoulder (Left) (Projection) Flex/Ext	0.518	0.284	0.002	0.003
Angle Shoulder (Right) Horizontal Rotation	0.506	0.279	0.007	0.002
Angle Shoulder (Right) Vertical Rotation	0.561	0.278	0.02	0.002
Angle Back Lateral Flexion	0.54	0.276	0.009	0.001
Angle Back Rotation	0.695	0.273	0.001	0
Angle Shoulder (Right) (Projection) Abd/Add	0.496	0.272	0.016	0.007
Angle Shoulder (Left) Rotation	0.537	0.27	0.004	0.001
Angle Shoulder (Right) (Projection) Rotation	0.486	0.255	0.003	0.001
Angle Neck Lateral Flexion	0.486	0.255	0.018	0.004
Angle Shoulder (Left) (Projection) Rotation	0.516	0.253	0.032	0.002
Angle Shoulder (Left) (Projection) Abd/Add	0.528	0.247	0.017	0
Angle Back Forward Flexion	0.528	0.247	0.081	0.012
Angle Shoulder (Right) Rotation	0.435	0.236	0.005	0.001
Angle Shoulder (Left) Horizontal Rotation	0.47	0.235	0.021	0.001
Angle Neck Rotation	0.402	0.224	0.007	0.001

Table 7. All features, sorted by greatest F1 score - UDT

Feature	Accuracy	F1	Acc Var	F1 Var
Angle Shoulder (Left) Horizontal Rotation	0.516	0.331	0.058	0.034
Angle Back Rotation	0.755	0.327	0.153	0.029
Angle Shoulder (Left) (Projection) Abd/Add	0.728	0.277	0.063	0.003
Angle Shoulder (Left) Vertical Rotation	0.569	0.27	0.075	0.004
Angle Neck Lateral Flexion	0.614	0.253	0.109	0.005
Angle Neck Flex/Ext	0.398	0.235	0.099	0.044

Angle Shoulder (Right) Horizontal Rotation	0.35	0.23	0.096	0.043
Angle Neck Rotation	0.564	0.226	0.151	0.013
Angle Shoulder (Right) (Projection) Rotation	0.341	0.221	0.089	0.036
Angle Shoulder (Right) Rotation	0.482	0.209	0.142	0.02
Angle Shoulder (Right) (Projection) Flex/Ext	0.291	0.195	0.036	0.018
Angle Back Forward Flexion	0.256	0.165	0.017	0.005
Angle Back Lateral Flexion	0.333	0.159	0.083	0.014
Angle Shoulder (Left) Rotation	0.3	0.157	0.081	0.014
Angle Shoulder (Left) (Projection) Flex/Ext	0.325	0.147	0.079	0.013
Angle Shoulder (Left) (Projection) Rotation	0.196	0.123	0.029	0.012
Angle Shoulder (Right) (Projection) Abd/Add	0.271	0.118	0.096	0.018
Angle Shoulder (Right) Vertical Rotation	0.125	0.079	0.015	0.005

Table 8. All features, sorted by greatest F1 score - SVM

Feature	Accuracy	F1	Acc Var	F1 Var
Angle Back Rotation	1	1	0	0
Angle Shoulder (Right) (Projection) Abd/Add	0.705	0.623	0.118	0.193
Angle Neck Rotation	0.678	0.61	0.181	0.219
Angle Neck Lateral Flexion	0.665	0.606	0.177	0.219
Angle Shoulder (Right) Vertical Rotation	0.662	0.603	0.189	0.225
Angle Shoulder (Left) (Projection) Abd/Add	0.678	0.513	0.097	0.112
Angle Shoulder (Right) Horizontal Rotation	0.471	0.405	0.147	0.169
Angle Shoulder (Left) Rotation	0.443	0.391	0.144	0.171
Angle Shoulder (Left) (Projection) Flex/Ext	0.444	0.391	0.142	0.169
Angle Neck Flex/Ext	0.485	0.388	0.152	0.171
Angle Shoulder (Left) (Projection) Rotation	0.483	0.386	0.154	0.173
Angle Back Lateral Flexion	0.461	0.386	0.142	0.17
Angle Shoulder (Left) Horizontal Rotation	0.459	0.384	0.149	0.175
Angle Shoulder (Left) Vertical Rotation	0.473	0.381	0.158	0.175
Angle Shoulder (Right) (Projection) Flex/Ext	0.423	0.368	0.155	0.18

Angle Back Forward Flexion	0.437	0.277	0.107	0.026
Angle Shoulder (Right) (Projection) Rotation	0.406	0.26	0.114	0.025
Angle Shoulder (Right) Rotation	0.391	0.253	0.107	0.025

Table 9. All features, sorted by greatest F1 score - PDT

Feature	Accuracy	F1	Acc Var	F1 Var
Angle Neck Flex/Ext*	0.384	0.257	0.022	0.015
Angle Back Forward Flexion*	0.521	0.229	0.085	0.011
Angle Back Lateral Flexion*	0.38	0.218	0.001	0.001
Angle Shoulder (Left) (Projection) Abd/Add	0.428	0.215	0.031	0.002
Angle Back Rotation*	0.488	0.197	0.111	0.018
Angle Shoulder (Left) Horizontal Rotation	0.342	0.18	0.012	0.001
Angle Neck Rotation	0.249	0.156	0.003	0.003

Table 10. All features, sorted by greatest F1 score - PD
* - value was reported using unbalanced training data

Bibliography

1. Hamilton-Wright, Andrew, and Daniel W. Stashuk. *Statistically Based Pattern Discovery Techniques for Biological Data Analysis*. pp. 3–31, https://doi.org/10.1007/978-3-540-78534-7_1.

2. scikit-learn. "1.10. Decision Trees," n.d. https://scikit-learn.org/stable/modules/tree.html.
   In-Text Citation: "1.10. Decision Trees."

3. scikit-learn. "1.4. Support Vector Machines," n.d. https://scikit-learn.org/stable/modules/svm.html.
   In-Text Citation: "1.4. Support Vector Machines."