University of Strathclyde Business School

DEPARTMENT OF MANAGEMENT SCIENCE

Smart Games

Data Analytics for Early Assessment in Autism

Poppy Harvey

September 2018

MSc Data Analytics

Except where explicitly stated, all the work in this dissertation – including any appendices – is my own and was carried out by me during my MSc course. It has not been submitted for assessment in any other context.

Signed:

Summary
It takes on average three and a half years from when a parent first raises concerns about their child before an official autism diagnosis is given in the UK. This diagnosis usually comes when the child is of primary school age and often past the point of when intervention can be most effective and provide the best prognosis.

The diagnosis process can be arduous and waiting times are long. Existing techniques focus on clinical observation of the child, and interviews with parents, guardians and caregivers. Clinicians are looking at how the child conducts themselves socially and emotionally to see if there is evidence of autism spectrum disorder. Autism, however, also effects how a person moves, and these differences in movements could be apparent long before social and emotional differences are observed.

This report documents the analysis of touch screen data, gathered from an iPad game played by children with autism (n = 37) and a control group (n = 45). Goal-oriented gestures made by the children during gameplay were identified, and significant differences were found in the speed, acceleration, jerk and duration of these gestures between the children with autism and their typically developing counterparts - lending further credence to the theory of the motor signature in autism.

This data was then used to train and test a variety of machine learning models with mixed results. While it was possible to build a model with recall in the autism class as high as 91%, precision suffered as a consequence. The best overall model was a Random Forest with an F_1 score of 0.68, an average precision of 0.68 and average recall 0.67.

This work contributes positively to a larger body of work looking at the viability of using Smart Games on tablet computers as a more time- and cost-effective alternative to the autism diagnosis status quo.

Acknowledgements

Thank you to everyone at the Laboratory for Innovation in Autism, for making this project possible and for making me feel so welcome, and especially to Szu-Ching, for all your help and guidance.

Thank you to Marc Roper for your advice and supervision, and for answering all my questions.

Thank you to my family for your encouragement all year, a special thanks to my mum for proof reading my report, and finally, thank you Steven, for all the millions of things you did for me to make the past few months a little bit less stressful.

Table of Contents

Project Context

Client Background

The Laboratory for Innovation in Autism at the University of Strathclyde is a research centre bringing together the disciplines of Electronic & Electrical Engineering, Biomedical Engineering, Psychology and Education.

Their research focusses on the theory of a motor disruption in autism spectrum disorder, with the aim of developing smart technology and wearable devices which can measure the movements made by children while they play. This information can then be used to characterise and better understand these motor differences and the effect they have on neurological development.

Their goal is that these wearables and smart devices can eventually be used as an accessible and fun alternative for the early assessment of autism and other neurodevelopmental disorders, in contrast to current practices which mostly involve lengthy observations and interviews, with long waiting lists.

The brief provided by the Laboratory was as follows:

"This Master's project will look at data from a set of experiments conducted by collaborators at the University of Southern California on iPad gameplay patterns to determine differences between children with autism, children with another neurodevelopmental disorder and children developing typically. Machine learning and bespoke data analytics methods will be applied."

Due to data availability, it was later decided that this project would only look at the differences between children with autism and children developing typically.

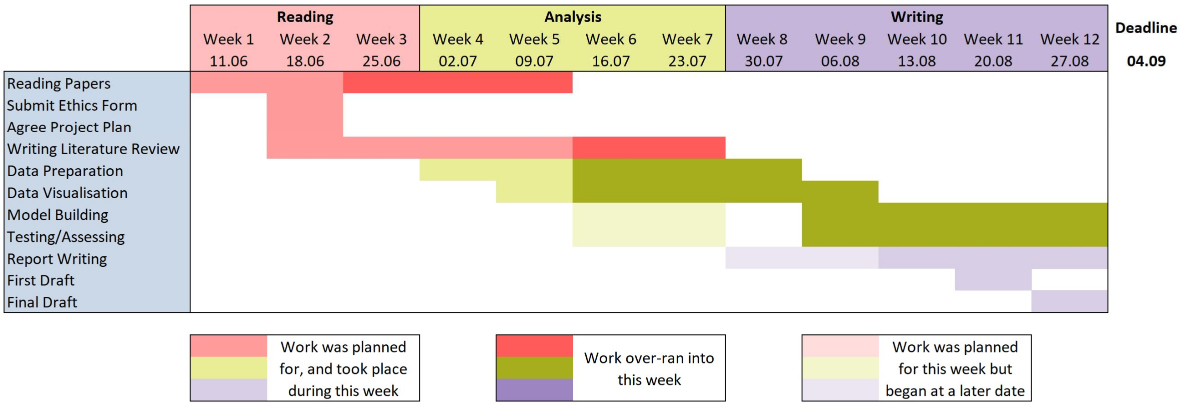
Research Questions

Building on this brief, and following discussions with both the client and project supervisor, a clearer vision of how the project would progress was formulated, finding a balance between what the client hoped to learn from the data and what was an appropriate scope and technical level for a Master's project.

Using various statistical, data analysis and machine learning methods, this project seeks to answer two main research questions:

- "Are there significant differences in the movements that children with autism make, compared to their typically developing counterparts, while interacting with an iPad game?"
- "Are these differences pronounced enough that a machine learning model can be trained to correctly classify the two cohorts of children?"

Project Plan



After discussion with both the client and project supervisor, it was agreed that the project would roughly follow a plan of reading and background preparations in June and analysis and evaluation in July, with August being left for writing up and contingency time. Subsequently this was developed into a more detailed plan encompassing the different stages of the project, shown above.

The data preparation and analysis took much longer than expected, due to the nature of the data, so having this contingency time was extremely useful. The literature review became a much more dynamic piece of work than expected, and while the majority of it had been completed as planned by the end of June, more papers were read during the data analysis process which were worth discussing and including.

In the end, the model building lasted right until the last week as various techniques were tried to improve the models. This was done in tandem with report writing, ensuring the project was delivered on schedule while making the most of the time available for undertaking the research.

DEPARTMENT OF MANAGEMENT SCIENCE

University of Strathclyde Business School

Smart Games

Data Analytics for Early Assessment in Autism

| Client Report

Poppy Harvey September 2018 MSc Data Analytics

Table of Contents

Table of Figures

1. Introduction

Autism spectrum disorder (ASD) is a developmental disorder that is estimated to affect around 1-2% of the population (Baird *et al.*, 2006; Baron-Cohen *et al.*, 2009; Brugha *et al.*, 2011; Kim *et al.*, 2011; Mattila *et al.*, 2011; Lai, Lombardo and Baron-Cohen, 2014; Russell *et al.*, 2014).

ASD is characterised by "persistent deficits in social communication and social interaction across multiple contexts" and "restricted, repetitive patterns of behaviour, interests, or activities" (American Psychiatric Association, 2013) and is normally diagnosed clinically, through behavioural observations (Lord and Volkmar, 2007). Tests and screening surveys such as the *Autism Diagnostic Observation Schedule* (ADOS)(Lord *et al.*, 2000), *Autism Diagnostic Interview-Revised* (ADI-R)(Lord, Rutter and Le Couteur, 1994), *Autism Spectrum Screening Questionnaire* (ASSQ)(Ehlers, Gillberg and Wing, 1999), *Modified Checklist for Autism in Toddlers* (M-CHAT)(Robins *et al.*, 2001) and the *Social Communication Questionnaire* (SCQ)(Rutter Bailey, A., & Lord, C., 2003) are often administered as part of this process.

It is generally accepted that an ASD diagnosis is possible from around two years of age (Brett *et al.*, 2016), and there is a wealth of research showing the benefits of early diagnosis and intervention for children with autism (Rogers and Vismara, 2008; Ben Itzchak and Zachor, 2011; Eikeseth *et al.*, 2012; MacDonald *et al.*, 2014) however diagnosis does not often occur until around primary school age (Shattuck *et al.*, 2009; Brett *et al.*, 2016), by which stage it may be too late for best intervention. A study by Goin-Kochel, Mackintosh and Myers (2006) found that over 40% of parents surveyed were not satisfied with the diagnostic process, and that satisfaction generally decreased as the number of professionals seen before getting a diagnosis increased.

There are myriad reasons for this delay in diagnosis, with factors such as socio-economic background, mother's age and education, birth order and geographic location all playing a part (Fountain, King and Bearman, 2011). Crane *et al.*, (2016) found that in the UK there was, on average, a delay of 3.5 years between parents first raising concerns about a child with healthcare professionals and getting an official diagnosis. Palmer *et al.*, (2011) surveyed all 243 UK child development teams listed on the British Academy of Community Child Health/British Academy of Childhood Disability database and found that of the 149 that responded only 36% had an official timescale within which to complete ASD assessments, and of these only 49% met the timescale recommended by the National Autism Plan for Children – less than 30 weeks.

If diagnosis could be simplified and timescales could be reduced, more children could reap the benefits of early intervention techniques, such as increased IQ, language skills, cognitive ability and reduced autism severity (Rogers and Vismara, 2008; Ben Itzchak and Zachor, 2011; Eikeseth *et al.*, 2012; MacDonald *et al.*, 2014).

This project set out to explore if a "Smart Game" solution could exist to aid in the early diagnosis of autism spectrum disorder – that is, could an iPad/tablet game aimed at young children be utilised to collect data, which a trained machine learning model can make predictions on, the result of which can then be used to inform clinicians whether the child playing the game is showing characteristics of autism?

This theory is based on extensive existing research showing that there is a difference between the gross and fine motor skills of children with autism and their typically developing counterparts. The iPad would collect *swipe gesture* data from children while they played a fun and engaging game, perhaps even unaware that they were being assessed. This theory relies on there being significant differences in how children with and without autism produce these gestures, such as the speed, acceleration and duration of the movement.

The research was undertaken with the support of the Laboratory for Innovation in Autism at the University of Strathclyde, who first proposed the research question and provided the data.

This report begins with a literature review examining the evidence for this motor difference, applications of machine learning in medical diagnosis and a look at other examples of experiments which analyse swipe gesture data collected from a smart device.

The methodology sets out how the experiment was conducted and data was collected, describes how the data was pre-processed and analysed and explains the motivation for the choices of the various machine learning models which were built, and the training and testing procedure which was undertaken.

The results from the initial data analysis and the machine learning can be found in the analysis section along with a commentary on their effectiveness, followed by a conclusions section which discusses how successful the experiment has been.

2. Literature Review

The literature review covers several topics, including an assessment of the status quo in autism spectrum disorder diagnosis, the feasibility of using touch screen devices with young children in a clinical setting, and an overview of the wealth of evidence for the motor signature in autism. It also includes examples of applications of machine learning for medical diagnosis, a specific look at where machine learning has been used in the context of diagnosing or further understanding autism, including experiments which analysed motor functions, and a broader look at research which involved touch screen swipe gestures and what can be inferred about the person making the gesture.

Current Autism Screening and Diagnostic Techniques

As mentioned in the introduction, autism is currently diagnosed through a series of clinical assessments which involve observation of the patient and, in the case of children, interviews with parents, guardians and teachers. These assessments follow a standardised procedure and each has its own strengths and weaknesses.

The ADOS is comprised of four modules, only one of which is administered - dependent on the age and verbal ability of the subject. The assessment requires the individual being observed to complete a series of tasks which involve social interaction with the examiner. It can be used in the diagnosis of autism and autism spectrum disorder in children from 12 months onward and takes between 30 and 60 minutes to complete (Lord *et al.*, 2000; Akshoomoff, Corsello and Schmidt, 2006; D. P. Wall *et al.*, 2012).

The ADI-R is an interview in which the caregivers of an individual suspected of having ASD are asked about the subject's behaviour, covering various topic areas in line with DSM-IV and ICM 10 diagnostic criteria (Lord and Volkmar, 2007). It is suitable for children, from the age of 18 months, and adults, and can take up to two and a half hours to complete (Lord, Rutter and Le Couteur, 1994; Dennis P. Wall *et al.*, 2012).

The ASSQ in not a diagnostic test but can be used for screening. The questionnaire takes around 10 minutes to administer, requires no training, and screening usually involves interviewing the child's parents/guardian and a teacher (Ehlers, Gillberg and Wing, 1999). The questions cover three topic areas – "social difficulties", "tics/motor/OCD" and "autistic style" (Mattila *et al.*, 2012). It is suitable for children between the ages of 7-16, however recent research suggests it could have validity with younger children (Adachi *et al.*, 2018).

Like the ASSQ, the M-CHAT is a screening questionnaire administered to parents of young children (Robins *et al.*, 2001). It takes less than 5 minutes to complete the 23 yes/no questions and can be used for children as young as 16 months (Hardy *et al.*, 2015). Neither the ASSQ nor the M-CHAT require trained professionals to administer, making them more accessible to parents, however under- and overreporting of symptoms by parents and guardians can reduce the validity of the results (Sipes and Matson, 2014).

The SCQ is another screening questionnaire designed to be answered by parents and that can be administered without the need for professional training. It consists of 40 yes/no questions and typically takes less than 10 minutes to complete. It was designed as a companion for the ADI-R and shares some characteristics with this, and is suitable for children above the age of 4 with a mental age of 2 years or more (Rutter, Bailey and Lord, 2003; Marvin *et al.*, 2017). Similarly to the ASSQ and M-Chat, the SCQ is vulnerable to under- or overreporting of behaviours by parents (Rudra *et al.*, 2014; Rubenstein *et al.*, 2017).

Using Touch Screen Devices for Screening or Diagnosis with Very Young Children

Research has shown that typically developing children are capable of using a touch screen device before their first birthday, with more than half of 3 year olds in one survey having their own tablet computer and between 71% and 97% of children having access to such devices in some capacity. One survey found that the median age for a child to be able to unlock a touch screen device, swipe across the screen and actively look for touch screen features was 24 months, and 25 months of age was the median age for children who could specifically identify and use touch screen features (Kabali *et al.*, 2015; Ahearne *et al.*, 2016).

A study into the feasibility of using touch screen technology for cognitive assessment in children was carried out (Twomey *et al.*, 2018) and found that typically developing children as young as 24 months were able to complete tasks on a tablet computer which required cognitive engagement; they were able to do this with minimal interaction with the administrator and no verbal instructions.

The weaknesses in traditional parent-based disability screening questionnaires, and the reliance on sometimes arduous clinical assessments were noted by the study's authors who state "formal testing is time consuming, requires expert assessors and therefore often is not available in clinics. A quick and reliable cognitive assessment tool that could be administered in clinics would allow for early identification of cognitive delay".

Evidence for the Motor Signature in Autism

While autism, and autism spectrum disorder, is most commonly typified by social and emotional impairment, narrow interests and a fondness for structure and repetition, there is huge body of evidence that autism can also affect a person's movements, with both fine and gross motor skills being affected in up to 79% of people with the condition (Green *et al.*, 2009; Lai, Lombardo and Baron-Cohen, 2014).

A literature review by Jennifer Cook (2016) presents a thorough summary of the research that has been carried out in the area of "movement kinematics and motor control in individuals with autism". She presents evidence that movements in autism are atypical - covering gait, posture, balance, fine motor control and upper limb movements. Referencing her own findings from earlier work (Cook, Blakemore and Press, 2013) on the topic of upper limb movements, she states "high functioning adults with autism make more jerky movements that proceed with greater acceleration and velocity, even when these movements are not goal directed and are thus relatively unconstrained.

Another literature review, Gowen and Hamilton (2013) deconstructs movements into a series of processes and takes a computational approach to reviewing the research relating to each area. In summarising their findings, they show many studies in which children with autism differed significantly from children in a control group.

A meta-analysis of research literature relating to motor deficits in autism spectrum disorder, which analysed data from 51 studies comparing motor functions in individuals with ASD and a control group, found significant diffences between the two cohorts and concluded motor coordination deficits to be a cardinal feature of autism spectrum disorder (Fournier *et al.*, 2010).

Bhat, Landa and Galloway (2011) present another body of evidence for motor delays and deficits in children and adults with autism and "propose that comprehensive motor evaluations are warranted for children with autism, regardless of age, and for infants at risk for ASDs" – such as siblings of those with an ASD diagnosis. They also suggest a link between motor and social impairments, hypothesising that a child without a "full movement repertoire" is hindered in social situations and that "poor coordination and slowed movement are linked to poor social participation and increased anxiety during playtime in the preschool and kindergarten years".

Brian *et al.* (2008) used ADOS and AOSI (Autism Observation Scale for Infants)(Bryson *et al.*, 2008) data collected for children with ASD, their infant siblings at 18 months, and a control group to identify behaviours that give the best indication at 18 months for a potential ASD diagnosis. Motor control information gathered from the AOSI test was found to be a reliable predictor for an autism diagnosis. 21% of children with ASD were found to have scored for motor impairment, compared to only 9% of non-ASD siblings and none of the control group. They state "The present findings highlight the importance of considering not only social-communication deficits when assessing for possible ASD in toddlers, but also basic dimensions of temperament, including state regulation, and possibly motor control and coordination, in achieving a more comprehensive assessment".

Trevarthen and Delafield-Butt (2013) hypothesised that, as there is a large body of evidence showing motor differences between typically developing (TD) children and children with autism spectrum disorder, and these motor differences can often be identified earlier than verbal or social differences, "examination of the psychobiology of motor affective disorders, rather than later developing cognitive or linguistic ones, may facilitate early diagnosis".

Machine Learning in Medical Diagnosis

On the topic of machine learning, and the increase of data being collected in the arena of healthcare, Maity and Das (2017) write "there is a dire need for data-driven approaches from computational sciences, often referred to as data science or data analytics, to help with understanding the data".

They discuss two examples of where machine learning techniques can be applied to interpret large volumes of medical data. For the first case they used a Bayesian model to diagnose Alzheimer's disease and were able to identify the Alzheimer's patients with 80% accuracy – however they do not state the ratio of Alzheimer's to non-Alzheimer's patients in the dataset and accuracy is a poor evaluation measurement for unbalanced classed. Their second case study involved classifying images of breast tissue cells into three possible early stages of cancer or benign. They built a neural network which classified the cell images with 90% accuracy – but again the distribution of images over the four classes was not reported in the paper.

Maity and Das conclude "Machine learning offers hope with early diagnosis of diseases, helps patients in making informed decisions on treatment options and can help in improving overall quality of their lives".

Machine Learning in Medicine (Deo, 2015) discusses some of the obstacles faced in translating the many examples of medicine based machine learning research into clinical settings, including the trade-off between accuracy and cost. A machine learning algorithm may be able to process medical data faster and at a lower cost than a doctor can, but is it worth it if the algorithm is not as accurate as the clinician, especially when it comes to people's lives? Thus, the need for extremely robust algorithms and models is apparent.

Another issue raised in this paper is the "black box" nature of many machine learning algorithms, also noted in (Sajda, 2006). Machine learning solutions are designed to work alongside doctors and other healthcare professionals, so it is important that these practitioners understand *how* these models work. This is more of an issue with neural networks than other machine learning techniques such as decision trees or linear regression but is important to consider.

Portability must also be considered when developing machine learning solutions for medical applications, especially diagnosis, as noted by Foster, Koprowski and Skufca (2014). Can the researcher's results be replicated in a clinical setting? How easy is it for healthcare professionals to gather the data to "feed" the algorithm? Importantly also, can the model be used by those without extensive statistical or data analytical knowledge?

Machine Learning and Autism

Autism is a heterogeneous disorder, and so it is unsurprising that the machine learning-based research into the condition is also.

Examples of research areas include MRI data (Lange *et al.*, 2010; Ingalhalikar *et al.*, 2011; Deshpande *et al.*, 2013; Sato *et al.*, 2013; Zhou, Yu and Duong, 2014), electroencephalogram data (Bosl *et al.*, 2011; Stahl *et al.*, 2012), eye movement data (Liu, Li and Yi, 2016), upper limb movement data (Crippa *et al.*, 2015), speech data (Xu *et al.*, 2009; Oller *et al.*, 2010; Motlagh, Moradi and Pouretemad, 2013), and ADOS and ADI-R data (D. P. Wall *et al.*, 2012; Dennis P. Wall *et al.*, 2012; Bone *et al.*, 2015, 2016; Kosmicki *et al.*, 2015).

Dennis Wall et al reviewed data from ADOS and ADI-R evaluations. Noting the long observation and waiting list times, they looked to find a minimum set of behaviours needed to accurately diagnose autism using these methods, in order to shorten the process.

In their first paper (D. P. Wall *et al.*, 2012a) they look at the Autism Diagnostic Observation Schedule - Generic and claim that their alternating decision tree was able to reduce the number of observation actions in module 1 (administered to pre-verbal) from 29 to 8 and still classify children with autism with 100% accuracy. Their second paper (Dennis P. Wall *et al.*, 2012b) concentrates on the Autism Diagnostic Interview – Revised. By again using an alternating decision tree model they claim they can classify children with autism and typically developing children with 99.9% accuracy using only 7 of the 93 questions contained in the ADI-R, which could drastically reduce the time needed to conduct the interview. A third paper (Kosmicki *et al.*, 2015) looks at modules 2 and 3 of the ADOS – module 2 is administered to children with limited vocabularies, module 3 is for children with fluent speech. They found that, once again using an alternating decision tree model, the ADOS module 2 could be reduced from 28 to 9 behaviours and classify children with 98.29% accuracy, and the model 3 reduced from 28 to 12 behaviours and still obtain 97.88% accuracy.

Their work was highly criticised by Bone *et al.* (2015), who failed to replicate the impressive accuracy figures claimed in (D. P. Wall *et al.*, 2012; Dennis P. Wall *et al.*, 2012). Criticisms include a misunderstanding in the design of the ADOS. By attempting to employ dimensionality reduction they ignore the fact that the ADOS scores are only reliable if the entire ADOS is administered – results will not be valid if only 8 of the 29 codes are scored on.

Secondly, in designing the analysis for both experiments Wall et al chose to remove individuals with a diagnosis in one of the middle diagnosis categories from both the ADOS and ADI-R leaving only the, more severe, autism diagnosis scores alongside the non-spectrum, or non-autism data – this makes the classification task simpler by only comparing two extremes, in reality ASD can be much more difficult to diagnose and accuracy of ~100% is unlikely to be achieved when including this data.

Uneven classes were also an issue in the methodology of Wall et al – their ADOS dataset included 612 individuals with an autism diagnosis and only 11 who were "non-spectrum", thus, even if one was to simply classify all subjects in the data as having autism they would achieve over 98% accuracy. Accuracy is a poor evaluation metric for uneven classes. In their ADI-R experiment their three data sets contained a total of 2867 subjects with autism and only 92 without.

Bone *et al.* (2015) conclude "computer scientists working in autism should be well versed in the autism literature, and autism researchers using machine learning should be confident in their understanding of these methodologies. Cross-fertilisation of this sort holds great potential for translational possibilities in in ASD research".

Machine Learning, Autism and the Motor Signal

Historically, handwriting analysis has been used to investigate fine motor control in subjects with autism. Johnson *et al.* (2013) conducted an experiment in which children aged between 8 and 13 wrote on a tablet computer using a stylus and found that the children with autism spectrum disorder had "significantly larger stroke height and width, more variable movement trajectory, and higher movement velocities" compared to their TD counterparts.

Building on their earlier work (Crippa *et al.*, 2013) in which they found evidence of inefficient eye-hand coordination in children with ASD, Crippa *et al.* (2015) investigated whether low-functioning children with ASD and an equal sized control group could be accurately classified using data collected during a reach, grasp and drop task.

Children were asked to pick up a rubber ball from a desk placed in front of them, and then place the ball into a box through a large opening. This was repeated 10 times, 5 using each hand, taking around 15 minutes to complete. The motion data was collected using a system of infrared motion analysis cameras and markers which were placed on the child's hand, the ball, and the box.

The researchers used a support vector machine model which was able to classify the two groups with a maximum accuracy score of 96.7% using seven data features, a maximum sensitivity score of 100% was achieved, meaning no children with ASD were missed. The 7 features all related to the part of the task which required the child to move the ball from its support to the target hole – "this suggests that goal-oriented movements may be critical in separating children with ASD from typically developing children".

This experiment showed that classification using motor data is possible, to a high degree of accuracy, however, despite having equal class sizes the sample size is small (30 children, 15 in each group), which can lead to overfitting of the model, it would be interesting to see if these results could be replicated on a larger cohort.

A strength of the task was that it was easy to understand, and all children were able to complete it, however portability could be an issue if this was to be used as a diagnostic aid, as it requires not only infrared motion analysis camera equipment but also the know-how to set these up and attach the markers to the correct areas of the hand and wrist.

In their paper, Anzulewicz, Sobota and Delafield-Butt (2016) or the Laboratory for Innovation in Autism at the University of Strathclyde, test whether or not it is possible to distinguish children with ASD from typically developing children by analysing their gesture interactions with two iPad games. They collected gameplay data from 37 children diagnosed with ASD and 45 children in the control group.

The first game, "Sharing", is a goal-oriented game requiring the child to slide pieces of food across a table to the plates of four animated children, when each child has been given a piece they will celebrate and a new item of food for sharing will appear – if a child does not receive food they will be visibly sad. The second game, "Creativity", was a simple colouring game in which children could choose one of a selection of simple line drawings to colour in, using their finger as if it were a crayon. This game had no rules and allowed for free play.

Using three decision tree-based machine learning algorithms and 262 features collected from both the touch screen sensors and inertial sensors of the iPad, they were able to classify the two groups of children with 84.8% accuracy from the sharing game and 92.6% accuracy from the creativity game.

The 10 features that contributed most to the classification from the Sharing game were all from the inertial sensor data – i.e. the movement of the iPad as a result of the force of the children's gestures, rather than the actual gestures themselves. For the Creativity game, 4 of the highest contributing features came from the inertial sensors and 6 from the touch screen interaction data. They found that the children with autism had greater impact force and gesture pressure, made larger, more distal gestures and had shorter screen tap duration (quicker taps). The impact force patterns and gesture pressure were also significantly different between the children with ASD and the control group - once again providing evidence of a motor signature in autism. It is this work that this project aims to build on, by seeing if these impressive classification results can be replicated, or improved upon, when only the gesture data is analysed, rather than the inertial sensor data which is sensitive to outside factors such as the child accidentally knocking the table on which the iPad is placed.

The work of Anzulewicz et al is not dissimilar to that of Crippa et al in theory, however the iPad-based set up offers an advantage over the motion sensor camera set up for a number of reasons. iPads are relatively cheap, accessible and portable. They are widely used outside of clinical settings, making them easier for medical professionals to employ, and less intimidating for children to interact with as they are likely to be familiar with them already. There is a much smaller margin for error with the setup for data capture compared to having to set up the motion sensor cameras and correctly position the markets.

Mahmoudi-Nejad, Moradi and Pouretemad (2017) also looked at whether children with ASD and typically developing children could be differentiated using data collected from an iPad game. Unlike Anzulewicz, Sobota and Delafield-Butt (2016), Mahmoudi-Nejad et al only looked at the touch screen data, and not gyroscopic data from the movement of the device, however they also included some in-game statistics in their analysis such as level completion scores.

Gameplay required the participant to trace their finger along a path of flowers towards a beehive. There were four levels of path difficulty, the easilest being a straight line and the most difficult comprising of some configuration of a spiral. There were 8 different paths at each level, which participants were prompted to trace twice, totally 16 "sub-levels" per level of difficulty.

The researchers found that the children with ASD were unable to complete levels about the easiest difficulty setting so only data from the first level was analysed for both groups. A total of 45 data features were captured, and forward feature selection was used to reduce this to 9 for use with a support vector machine classifier. The data from each child's 16 "sub-level" plays was averaged to provide one data profile per subject.

The SVM model achieved 100% accuracy in classifying the children with ASD and the TD children, and claims to have only needed two data features to achieve this, stating "these children [children with ASD diagnosis] have low accuracy in following a given trajectory". It is, however, important to note that the sample size was only 12 (5 ASD, 7 TD) which is extremely small, suggesting over-fitting of the model could have been an issue. Another weakness of this investigation is that no female children were included in the ASD group.

"Swipe" Gesture Analysis

Touch screen gesture data was analysed by Bevan and Fraser (2016) with the aim of finding a relationship between thumb length and swipe characteristics – hypothesising that thumb length gives an indication of arm length which in turn gives information about a person's height that could eventually be used as a "soft biometric" for identification.

Data was collected using a purpose built smartphone app which prompted users to swipe either up, down, left or right to reveal the punch line of a simple joke, using just their thumb (left or right depending on preference/handedness). 21,360 swipe gestures were collected from 178 adult participants (87 male, 91 female) between the ages of 18 and 59. The thumb lengths of all subjects were also measured, using the same defined start and end point for everyone.

Using various traditional statistical techniques, the researchers found that participants with longer thumbs produced swipes with shorter duration and higher speed and acceleration.

This work was built on by Miguel-Hurtado *et al.* (2016), in which various machine learning classifiers were employed to predict gender using the swipe data collected using the same method as in (Bevan and Fraser, 2016). Data features included the length of the swipe, the time taken to complete the swipe, maximum and average speed and acceleration and the average pressure exerted on the screen during the swipe.

At first the swipes were broken down into direction and analysed – left-to-right, right-to-left, up-to-down and down-to-up, with equal number of each swipe having been collected. A Naïve Bayes classifier produced the highest accuracy for any swipe direction – 71.8% of left-to-right swipe gestures were classified correctly. A fusion technique was used to combine the predictions for all swipe directions for each user, and this increased the accuracy to 78.2% using a decision voting scheme.

Conclusion

Waiting times for autism diagnoses in children are long, many children are missing out on early intervention and the benefits that come with this, and the majority of parents are not satisfied with the current process. A screening or diagnosis tool that could shorten this timescale could offer a better prognosis for many children.

Decades of research has shown that autism does not only have social and emotional effects, despite these being the most well-known characteristics of the disorder. There are significant differences between the fine and gross motor skills of many people with autism spectrum disorder and the general population.

Evidence shows that children are able to interact with iPads and other smart, touch screen devices from an increasingly younger age. These devices are also relatively cheap, portable, and familiar to most clinical practitioners.

If the significant motor differences in autism could be captured by way of a touch screen device, machine learning could be used to build a classifier to be used in the diagnosis of ASD, which may be faster, and less susceptible to bias than the existing observational and interview methods which are currently used.

3. Methodology

This chapter describes the dataset which was used in this data analysis project, how the data was collected, the pre-processing that was undertaken to prepare the data for modelling, the building of various machine learning models, and the training and testing of these models.

3.1. Data Collection and The Dataset

The dataset for this project was provided by the Laboratory for Innovation in Autism at the University of Strathclyde. Touch screen data from an iPad mini was recorded while children interacted with a game called "Sharing". 82 children took part in the study, 37 of these children, with an average age of 4 years and 5 months, had a diagnosis of childhood autism, while the control group was comprised of 45 typically developing children with an average age of 4 years and 7 months.

All children had normal or corrected-to-normal vision and no other motor or sensory defects. Of the 37 children with the childhood autism diagnosis, 30 were considered uncomplicated, two had an Asperger's Syndrome diagnosis, one was "high functioning" and four had an additional diagnosis of "intellectual impairment".

The "Sharing" game, Figure 1, required the children to tap on a piece of food in the centre of a table on the screen, which would split the food into four parts, and then drag each of these parts on to the plates of four animated children who were seated at the table. Once each child had received a piece they would celebrate, the game would reset and a new food item would appear. If the food was distributed unevenly, or not all children received a piece of food, the game would not continue until this was rectified and the children without food would display sad expressions. "Distractor elements" were also present in the game, such as ceiling lights which could be switched on or off with a tap.

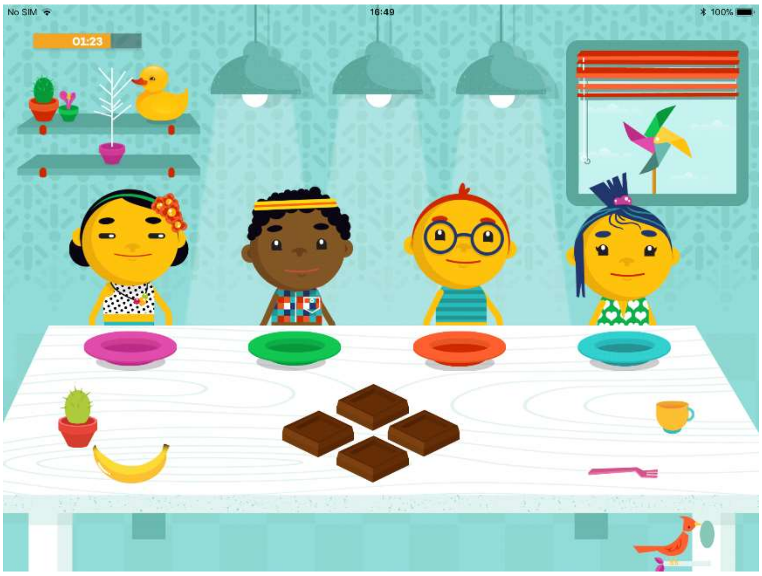


Figure 1: Screenshot of the sharing game

The iPad was contained within a rubber/silicone-type case which was secured to the surface of a children's table with double-sided tape. Each session began with a two-minute tutorial, during which the main gameplay aims were highlighted with animations on the iPad screen and an experimenter explained the game both verbally and by using the device to demonstrate to the child. This was followed by five minutes of gameplay during which time the examiner did not interact with the game. The child's teacher or clinician was present throughout.

The touch screen data was recorded at a frequency of approximately 60Hz (fluctuating slightly depending on the efficiency of the iPad), and was uploaded to a cloud storage service after each session. The dataset provided by the Laboratory for Innovation in Autism contained each of the 82 children's gameplay sessions data in separate CSV files. Whether the child belonged to the ASD cohort or the control group was indicated in the filename.

The data features contained in each CSV file are explained in Table 1 below:

Feature Name	Data Type	Description
sessionID	String	A unique identifier for each gameplay session – can be used as a proxy identifier for each subject as all children only played one session
x	Float	The x co-ordinate of where the screen was touched
y	Float	The y co-ordinate of where the screen was touched
touchPhase	Integer	Takes one of five values 0 *Began* – a finger for a given event touched the screen 1 *Moved* – a finger for a given event moved on the screen 2 *Stationary* – a finger is touching the screen but hasn't moved since the previous event 3 *Ended* – a finger for a given event was lifted from the screen 4 *Cancelled* – the system cancelled tracking for the touch, as when (for example) the user moves the device against his or her face *Source:* https://developer.apple.com/documentation/uikit/uitouch/phase
tapCount	Integer	The number of times the finger was tapped for this given touch
touchNumber	Integer	The number of fingers touching the screen at one time
time	Float	The time at which the screen touch was recorded – the clock does not reset to zero for each gameplay session so time values are only relevant relative to other timestamps from within the same session

Table 1: Data Features of the Touch Screen Dataset

3.2. Software

Analysis was carried out using Python 3.6 and Spyder (The Scientific PYthon Development EnviRonment) version 3.2.8, with some visual inspection of the data carried out using Microsoft Excel (2016).

3.3. Data Pre-Processing

Visualising the Dataset

Each of the 82 CSV files containing data from the subjects' gameplay sessions were read into the Spyder environment, and were concatenated into one large DataFrame. A new feature indicating whether the child had an ASD diagnosis (Label = 1) or not (Label = 0) was added to the DataFrame and the data was sorted by time.

Figure 2 and Figure 3 show the value distributions for the tapCount, and touchNumber variables respectively. As can be seen from the histogram, the touchNumber variable only ever takes the value 0, so it was discarded from the dataset. The tapCount variable was also removed as it also almost exclusively took the value 1 and the focus of this analysis is "swipe" gestures rather than taps, so it is of negligible value.

Figure 4 shows the frequency counts for the touchPhase feature. Case 2 (indicating a stationary finger on the screen) never occurs in this dataset, and case 4 (a cancelled gesture) occurred only 7 times (1.4×10^{-5}% of all recorded touchPhase values). The vast majority of cases were 1 – the finger is moving on the screen (completing a "swipe" gesture), with roughly equal numbers of 0s (swipe begins) and 3s (swipe ends), as would be expected.

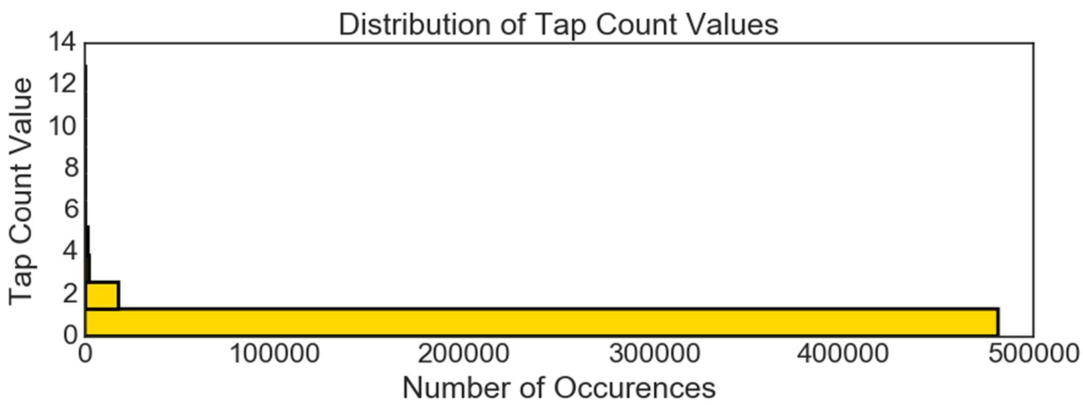


Figure 2: Distribution of Tap Count Values

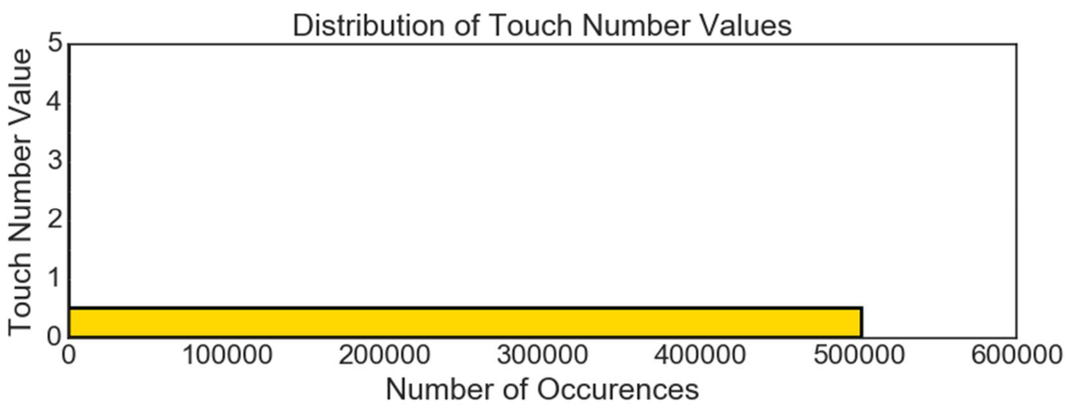


Figure 3: Distribution of Touch Number Values

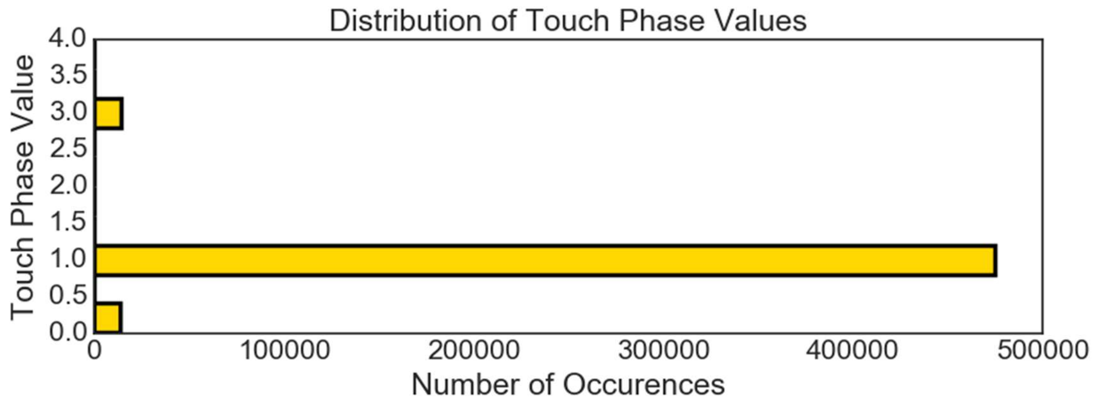


Figure 4: Distribution of Touch Phase Values

Figure 5 and Figure 6 show heatmaps of where the children with an autism diagnosis and the children in the control group touched the iPad. Figure 5 is much noisier than Figure 6– the typically developing children mostly concentrated on playing the game - sharing the food, whereas some of the children with ASD were more likely to interact with the game's "distractor elements" such as the ceiling lights, cactus and bird – or simply touch the screen indiscriminately.

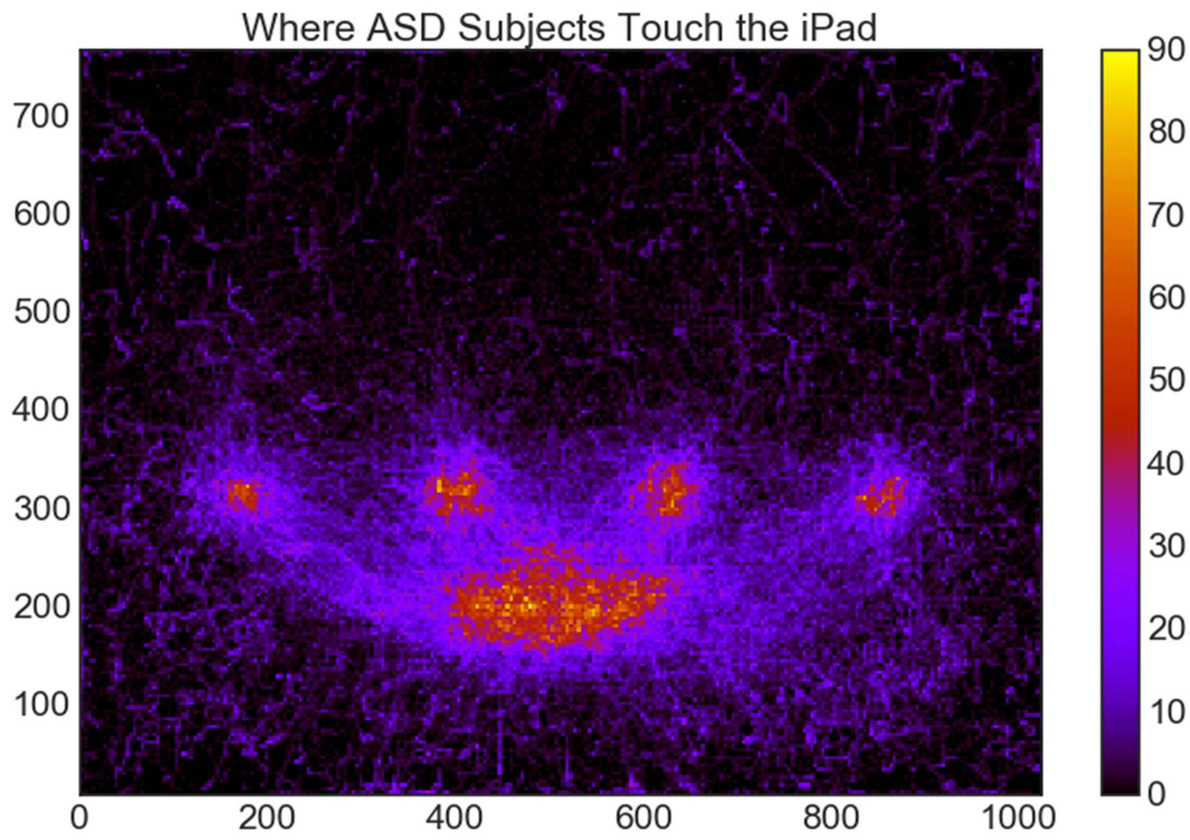


Figure 5: Heatmap of where children with an ASD diagnosis touched the iPad

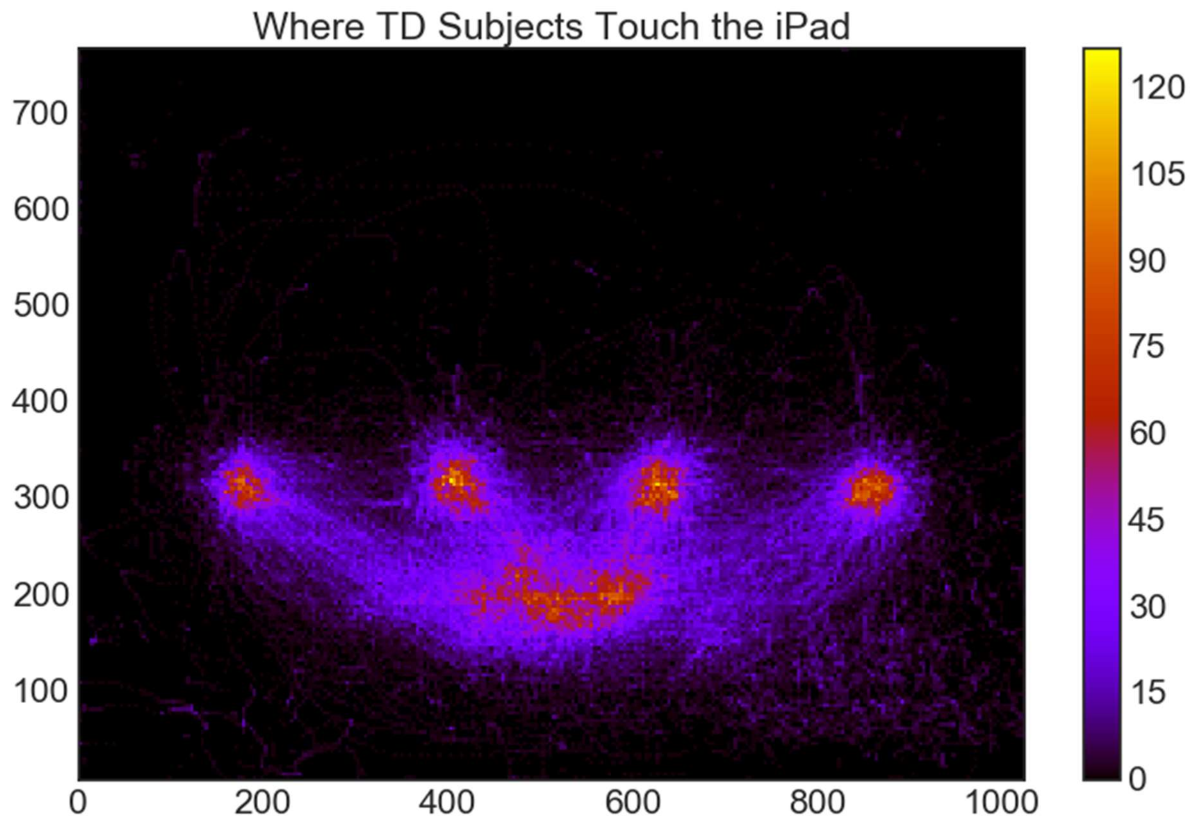


Figure 6: Heatmap of where typically developing children touched the iPad

Identifying Swipe Gestures

As described in Table 1, the beginning of a gesture is indicated by touchPhase = 0, and the end by touchPhase = 3 or, very rarely, 4. Visual inspection of the touchPhase variable showed that not all gestures were perfectly defined. To the left of Figure 7 is an example of "clean" data – the first entry in the touchPhase column is a 0 – the beginning of a swipe gesture, followed by a series of 1s and concluding with a 3 – a distinct swipe. In contrast, on the right of the figure, is an example of "unclean" data. After the first swipe ends there should be another 0 to indicate the start of the next swipe, however there are a series of 1s instead, then a short swipe or tap, followed once again by a series of 1s in the touchPhase column. These errant 1s needed to be excluded from the dataset before further analysis could be carried out.

A temporary DataFrame was created which contained only the rows where the touch phase was either 0 or 3. By iterating through each row of this DataFrame looking for 0s and checking for a 3 in the next row's touchPhase column, 11438 completed gestures were identified. With the beginning and end points of these gestures known, all touch phase values equal to 1 that fell between these parameters were kept and all other datapoints were removed from the dataset. A new column was added to the DataFrame and each gesture was assigned a number to act as an identifier.

sessionId	x	y	touchPhase	time
76757CF5-	226	352	0	42111.75
76757CF5-	215.5	358.5	1	42111.78
76757CF5-	211	358.5	1	42111.80
76757CF5-	208.5	358.5	1	42111.82
76757CF5-	206.5	358.5	1	42111.83
76757CF5-	205	359	1	42111.85
76757CF5-	204	359	1	42111.87
76757CF5-	202.5	359.5	1	42111.88
76757CF5-	201.5	359.5	1	42111.90
76757CF5-	200	360	1	42111.92
76757CF5-	198.5	360	1	42111.93
76757CF5-	196	360.5	1	42111.95
76757CF5-	193.5	361.5	1	42111.97
76757CF5-	190	362	1	42111.98
76757CF5-	187	363	1	42112.00
76757CF5-	183.5	364	1	42112.02
76757CF5-	180.5	365	1	42112.03
76757CF5-	178	365.5	1	42112.05
76757CF5-	175.5	365.5	1	42112.07
76757CF5-	173	365.5	1	42112.08
76757CF5-	170	366	1	42112.10
76757CF5-	167	366	1	42112.12
76757CF5-	164.5	366	1	42112.13
76757CF5-	162.5	366	1	42112.15
76757CF5-	160.5	366	1	42112.17
76757CF5-	159	366	1	42112.18
76757CF5-	157	366.5	3	42112.20

sessionId	x	y	touchPhase	time
5CD175F9-	891	325	0	10682.34
5CD175F9-	757.5	508	1	10682.56
5CD175F9-	1001.5	595.5	1	10682.17
5CD175F9-	830	726	1	10681.62
5CD175F9-	654	623	1	10682.66
5CD175F9-	660	672	1	10681.94
5CD175F9-	731.5	592	1	10682.71
5CD175F9-	982.5	561	1	10683.57
5CD175F9-	870.5	339	1	10682.62
5CD175F9-	927.5	367	1	10681.67
5CD175F9-	675	619	3	10681.77
5CD175F9-	1006	616	1	10682.01
5CD175F9-	589	344	1	10682.62
5CD175F9-	862	505.5	1	10682.27
5CD175F9-	743	252	0	10681.52
5CD175F9-	740.5	222.5	1	10681.96
5CD175F9-	908.5	353	1	10681.92
5CD175F9-	746.5	188.5	3	10682.79
5CD175F9-	730	421	1	10681.94
5CD175F9-	684.5	523	1	10681.82
5CD175F9-	741.5	247	1	10681.56
5CD175F9-	716	718.5	1	10682.09
5CD175F9-	1019.5	652.5	1	10681.54
5CD175F9-	685	581	3	10681.71
5CD175F9-	661	672.5	1	10681.92
5CD175F9-	711	713	1	10682.9
5CD175F9-	677.5	669	1	10682.27

Figure 7: Examples of "Clean" and "Unclean" Data

Figure 8 shows the lengths of these gestures, measured in recorded touch phases, logically the shortest gestures last for two touch phases – a 0 and a 3, i.e. a "tap" on the screen. Some tap-like gestures may last longer than this depending on how slowly the child released their finger. The touch points were plotted using the x and y co-ordinates and, by visually inspecting these plots, any gesture lasting for less than 10 touch phases was deemed to be a tap and removed from analysis. Removal of these tap gestures left 8000 swipe gestures in the dataset. While the longer swipes – lasting up to 431 touch phases, may seem erroneous, visual inspection of the data showed them to be legitimate gestures.

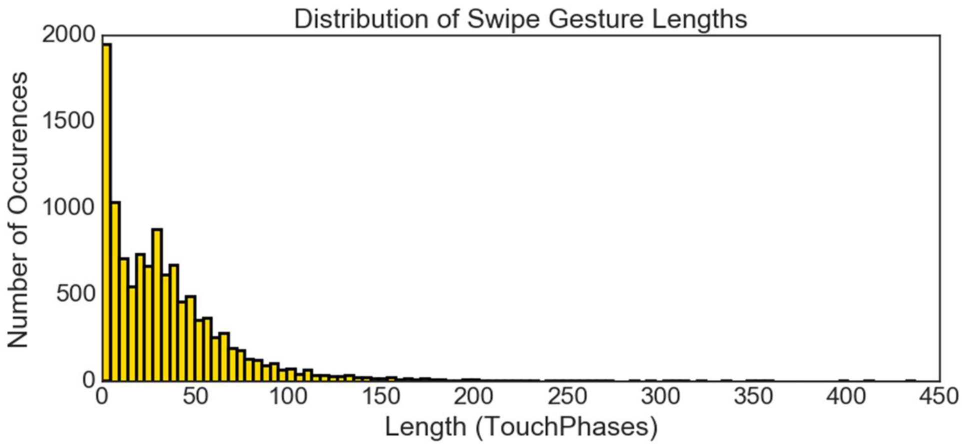


Figure 8: Lengths of the identified gestures, by number of recorded touch phases

The settings within the Sharing game allowed for "multitouch" gestures – i.e. the iPad would record multiple touch occurrences with the same timestamp if a child touch the iPad with more than one finger at a given time. These gestures were discarded as this project aimed only to analyse single-finger movements – bringing the number of gestures down to 6661.

Finally, goal-oriented gestures were identified – i.e. those gestures where the child is dragging the food from the centre of the table to each of the plates, herein referred to as "Food-to-Plate" gestures. Some of the subjects were inclined to drag the food to the animated children's mouths, and during gameplay food delivered in this fashion would "snap" to the plate and count as a success, so these gestures were also included in analysis, referred to as "Food-to-Face". A total of 4757 goal-oriented gestures were identified in the data, the breakdown of this, by cohort and gesture type, can be seen in Figure 9. Interestingly, the proportion of goal-oriented gestures which were Food-to-Face swipes (rather than Food-to-Plate) for the two groups of children was almost equal – ASD: 31.6%, TD: 31.9%.

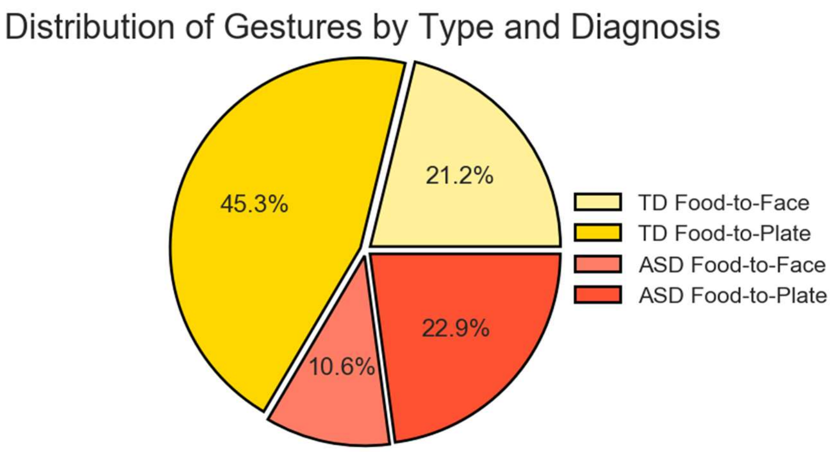
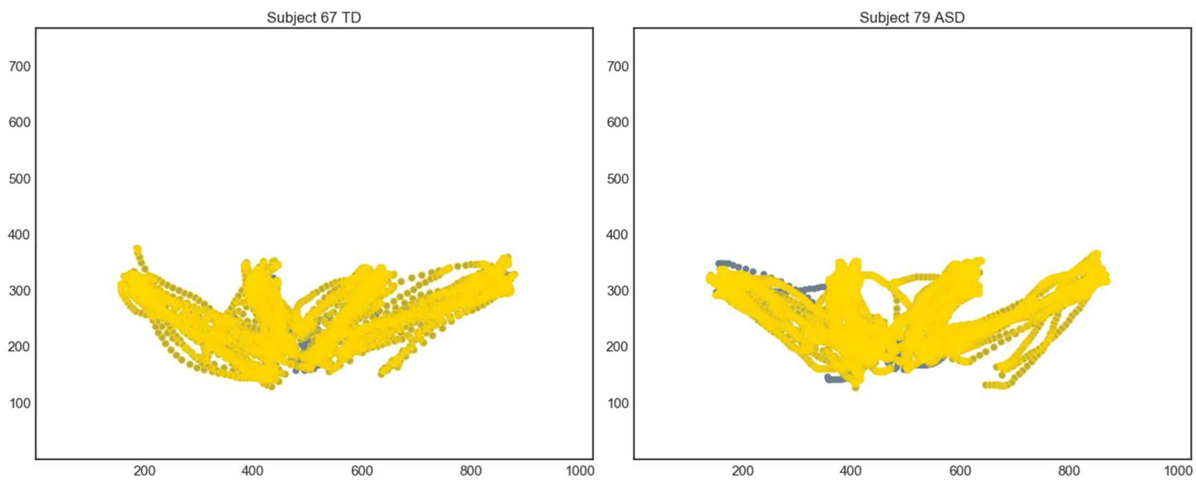


Figure 9: Proportion of gestures by cohort and gesture destination

Figure 10 shows the identified goal-oriented swipes in yellow, and all other recorded touch points in grey, for a selection of subjects from the study. The top two panels show the data collected from a child in the control group – on the left, and a child with an autism diagnosis – on the right, both of whom



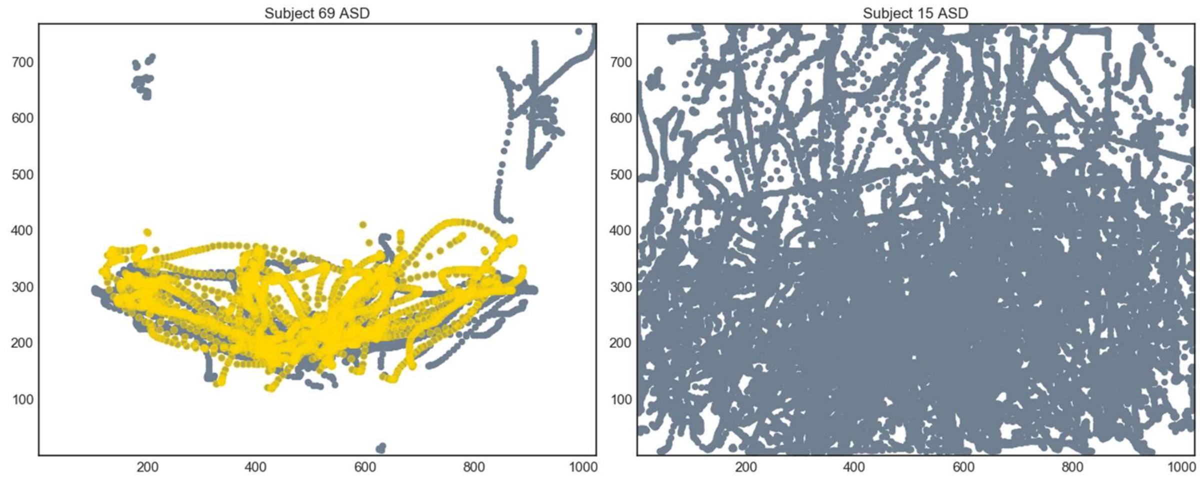


Figure 10: Examples of swipe gestures from four subjects

played the Sharing game as intended and made goal-oriented gestures almost exclusively. The bottom left panel shows another child from the ASD cohort who made many goal-oriented, food-sharing gestures but also interacted with some of the "distractor elements" in the game – such as the rubber duck and window blind which can be seen in *Figure 1*. Finally, the bottom right panel shows the data collected from a child in the ASD group who completed no Food-to-Plate or Food-to-Face swipes and appears to have interacted with the iPad in a completely random manner.

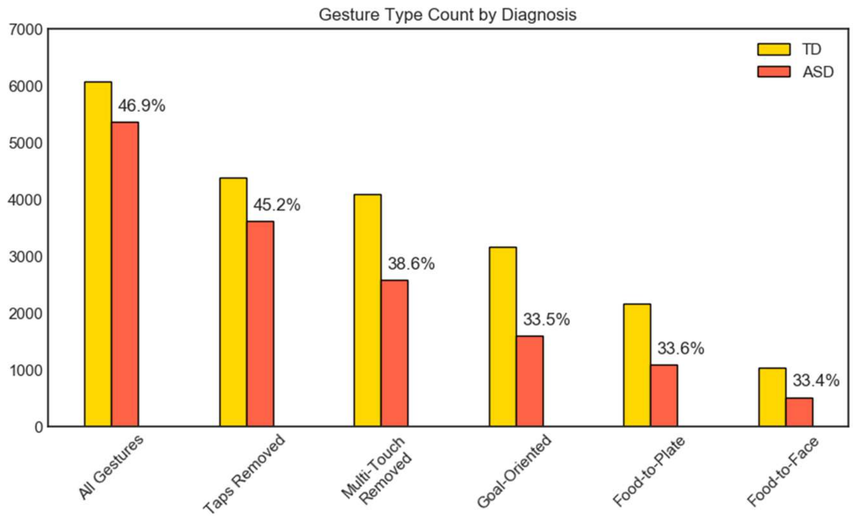


Figure 11: Number of recorded gestures at each stage of the data preparation process, grouped by cohort

Figure 11 shows the number of gestures for each cohort at every stage of the data preparation process, with the percentage of these gestures which were made by children in the ASD group shown above each bar.

At the first stage, when any gesture with a defined start and end point was included in the DataFrame, the ASD gestures made up 46.9% of the total – this is slightly higher than the representation of children with ASD within the experiments (37 out of 82 – 45.1%) – suggesting they made more gestures on average than the TD children.

When taps are removed the distribution of gestures between the cohorts is almost the same as the distribution of participants across the two groups. It can be seen however than the more the data is cleaned – removing multi-touch gestures and gestures which are not goal-oriented, the less the children with autism are represented in the dataset. The final data set of gestures being taken forward for further analysis features almost twice as many swipe gestures from typically developing children than gestures from children diagnosed with autism.

3.4. Kinematic Measurements

Kinematics refers to the analysis of motion, without reference to the forces which cause the motions or the masses of the objects involved.

Speed, also known as the magnitude of velocity, is the first time derivative of position, i.e. the rate of change of position P - Equation 1. Position in this case is defined on the x-y plane as the iPad screen is two dimensional. Acceleration is the second time derivative of position, the rate of change of speed v - Equation 2. Jerk is the third time derivation of position, or the second time derivative of speed, i.e. the rate of change of acceleration a - Equation 3. This analysis is only concerned with the magnitude of these values, not direction, and thus all values are scalar.

$$Speed\ (v) = \frac{\Delta P}{\Delta t}$$

Equation 1

$$Acceleration\ (a) = \frac{\Delta v}{\Delta t}$$

Equation 2

$$Jerk\ (j) = \frac{\Delta a}{\Delta t}$$

Equation 3

As mentioned in the literature review, various experiments have been carried out which investigate upper body movements and fine motor skills, looking to find evidence of the motor signature in autism. It is these kinematic measurements which are often used to make comparisons between study participants with ASD and a control group, and to look for significant differences in their motor functions.

Cook, Blakemore and Press (2013) conducted an experiment in which subjects were recorded, using infrared cameras, making back-and-forth arm movements. Analysis showed that the movements from subjects in the autism group had significantly higher velocity, acceleration and jerk values than their control counterparts. The movements of those in the autism group also had significantly shorter durations and greater distances thank those in the control group.

Glazebrook, Elliott and Lyons (2006) conducted an experiment in which adults with autism, and a control group, were recorded moving their finger between a series of targets which were projected onto a board in front of them. Their analysis found that the participants with autism took significantly longer to execute their movements than those in the control group, and that the control group had a significantly higher mean peak velocity and mean peak acceleration than the ASD group.

Forti *et al.* (2011) carried out a reach-to-drop task with preschool aged children, where participants were instructed to pick up a rubber ball and then drop it into a box through a hole. They found that movement duration was significantly longer for those in the autism group, however while the peak velocity was higher for the control group it was not significantly so.

A reach-to-grasp experiment was carried out by Mari *et al.* (2003), in which primary school-aged children with autism were asked to reach across a table and pick up a small Perspex box. Their research found that there were substantial differences in the kinematic parameters between children in the autism group depending on if their IQ was categorised as being "low" or "medium to high". The low IQ subjects with an autism diagnosis had significantly longer movement durations and performed the task with significantly lower peak velocity than the control group, however the medium to high IQ children with autism performed the task with significantly higher peak velocity than the control group and had significantly shorter movement durations.

Calculations

To analyse the gesture data from the Sharing game, *speed*, *acceleration* and *jerk* values have been calculated for each completed, goal-oriented gesture, along with the *distance* and *duration* of these movements.

As described in Table 1, the dataset contained x and y co-ordinates for each touch phase data point, i.e. position *P,* alongside a timestamp, *t*. Figure 12 shows an illustrative example of a typical swipe gesture, with the locations of touch phase readings shown as circles – thus each circle represents a row in the data set, with an accompanying x co-ordinate, y-co-ordinate, touch phase (0, 1, 3 or 4) and timestamp.

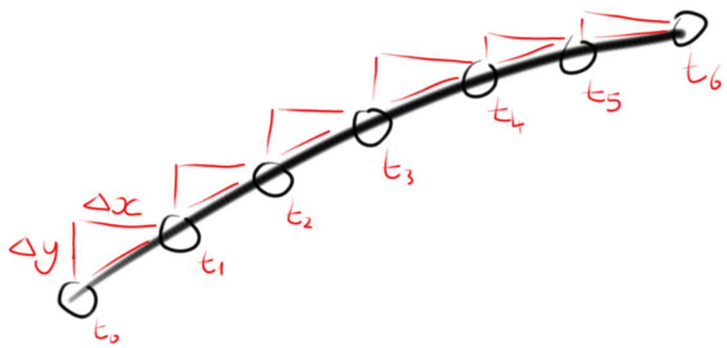


Figure 12: An illustrative example of a swipe gesture

The duration of each swipe was calculated by subtracting the timestamp at the start of the gesture (touchPhase = 0) from the timestamp at the end of the gesture (touchPhase = 3 or 4). The distance of the swipe was calculated as the sum of the distances (Euclidean norm) between each touch phase reading - Equation 4, where *n* is 1 less than the number of touch phases recorded for a gesture. Speed

v, for each swipe gesture was then calculated by dividing distance by duration. For each swipe a value of minimum, mean and maximum speed was recorded.

$$Distance = \sum_{i=0}^{n} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

Equation 4

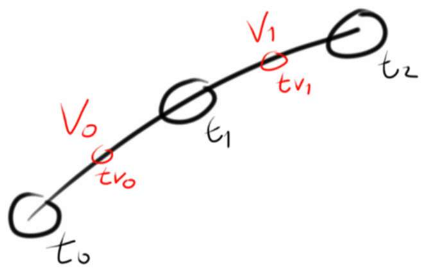


Figure 13: Calculating timestamps, tv, for the speed values, v

To calculate the acceleration of each swipe it was necessary to compute a time *tv*, to accompany the speed values which had been calculated. This was taken as the average of the two timestamps used to compute the duration for the speed calculations -Equation 5, see Figure 13. These new time values were then used in the acceleration calculations -Equation 6. Minimum, mean and maximum absolute acceleration was recorded for each swipe gesture.

$$tv_i = \frac{t_i + t_{i+1}}{2}$$

Equation 5

$$a_i = \frac{v_{i+1} - v_i}{tv_{i+1} - tv_i}$$

Equation 6

Jerk calculations were carried out in a similar fashion - Equation 8, with timestamps for the acceleration values, ta_i, being calculated as the average of tv_i and tv_{i+1} - Equation 7. Minimum, mean and maximum absolute jerk were recorded for each swipe gesture.

$$ta_i = \frac{tv_i + tv_{i+1}}{2}$$

Equation 7

$$j_i = \frac{a_{i+1} - a_i}{ta_{i+1} - ta_i}$$

Equation 8

3.5. Machine Learning Models

The overall aim of this project was to see if it is possible to correctly determine if a child has autism spectrum disorder or not, based on how they interact with the Sharing game on the iPad. This is an example of a *binary classification problem*, the child either belongs to the ASD group, label = 1, or the typically developing control group, label = 0.

Decision Trees

A Decision Tree is a simple, yet power, type of machine learning model. It works by splitting the data using a series of comparison operators on its features. It is known as a *white-box* model as it is possible to see which feature values led the classifier to make its predictions on each data instance.

Another major benefit of the Decision Tree model is that is requires very little preparation of the data, such as feature scaling, however a drawback of this model type is that it can be sensitive to small changes in the training data.

The *DecisionTreeClassifier* function within Scikit-Learn for Python uses the CART (Classification and Regression Tree) algorithm to split the data. This method begins by splitting the data into two *nodes* using the feature which produces the "purest" subsets, measured by the *gini score*, weighted by their size. Purity refers to the spread of classes which training instances in the node belong to. The purest node (gini = 0) will have training instances belonging to only one class. The algorithm uses the same principle to then split the subsequent nodes until either a pre-defined maximum depth is reached or no further splits will reduce the impurity score.

Ensemble Learning

Ensemble learning refers to the process of aggregating the predictions of a group of classifiers (or regressors), which often leads to a model that produces better results.

One example of an ensemble learning method is Bootstrap Aggregating, or more commonly *Bagging.* This involves building several predictors using the same algorithm but training them on different random subsets of the training set. In this method, training samples are replaced into the pool of potential samples meaning a predictor may be trained on the same data instance several times. When data instances are not replaced in the training set this method is knowing as *Pasting*.

Predictions are made by the ensemble using *hard voting* i.e. the model uses the most commonly occurring prediction given by the predictors. Using the Bagging method results in a model with lower variance than an individual predictor trained on the whole training set.

With Bagging it is possible that not all training instances will be "seen" by the model as training samples are replaced and thus some are seen more than once. This unseen data can be used to evaluate the model, much like a test set, and produces a metric known as the *out-of-bag* score.

Random Forests

Random Forests are a very powerful type of ensemble method which use a bagging technique which is, as its name may suggest, optimised for Decision Trees. The "random" aspect is introduced when the trees in the ensemble are splitting nodes. At each node the model must choose the best feature on which to split from a random subset of features, rather than all features, which increases the diversity of the trees in the ensemble. This lowers the variance of the model which generally produces better results than a Bagging classifier.

Support Vector Machines

Another very popular, and powerful, machine learning model is the Support Vector Machine (SVM). The simplest SVM will separate a two-class, linearly separable dataset by creating a hyperplane between the two classes of data, with the largest possible margin between the closest data instances in each class – called support vectors. This is known as linear classification.

When the data is not linearly separable, *soft-margin* classification is used. This involves finding a balance between the width of the margin and the number of *margin violations* – data points that fall within the margin, including those which fall on the incorrect side of the hyperplane. This fine tuning is possible within the Scikit-Learn SVM function for Python by adjusting the *C* hyperparameter.

An alternative method for non-linearly separable datasets is to use a *kernel trick*. This gives the effect of transforming the data using some function, which increases the chance of it being linearly separable, without actually having to create the additional data features – saving time and computational energy. Examples of kernels that can be used are polynomial, which simulates adding polynomial features to the data, and the Gaussian Radial Basis Function, which computes similarity measures for each data point based on *landmarks* that it defines.

All three of these Support Vector Machine methods are easy to implement and tune using the Scikit-Learn package for Python.

3.6. Building, Training and Testing the Models

Decision Tree, Bagging, Random Forest and Support Vector Machine models were all built, trained and tested on the Sharing game data. Each model has its own set of hyperparameters which can be varied to find the best fit. The *GridSearchCV* function was used to find the optimal values for these parameters for each model.

The final data features used for training and testing the models, and a short description of each, are listed in Table 2. The data was split, with 90% being used for training and the remaining 10% for testing. As Figure 9 shows, there are twice as many swipe gestures in the TD class as there are in the ASD class. This is an example of unbalanced classes, which is not ideal when training machine learning models as often the model does not have enough information to learn about the minority class and can subsequently heavily over-predict the majority class, resulting in a poor predictor.

This problem was overcome by *oversampling* the training set – the *RandomOverSampler* function does this by duplicating data instances from the minority class until both classes are of equal size. Oversampling was not carried out on the test set as in a real-world environment the class sizes are unlikely to be even, and this could result in the evaluation metrics of the model being unrealistic.

Data Features	
Minimum Speed	Minimum speed recorded during the swipe movement
Average Speed	Average of the recorded speeds during the swipe movement
Maximum Speed	Maximum speed recorded during the swipe movement
Minimum Acceleration	Minimum acceleration recorded during the swipe movement
Average Acceleration	Average of the recorded accelerations during the swipe movement
Maximum Acceleration	Maximum acceleration recorded during the swipe movement
Minimum Jerk	Minimum jerk recorded during the swipe movement
Average Jerk	Average of the recorded jerks during the swipe movement
Maximum Jerk	Maximum jerk recorded during the swipe movement
X_1	The initial x co-ordinate of the swipe movement
X_2	The final x co-ordinate of the swipe movement
Y_1	The initial y co-ordinate of the swipe movement
Y_2	The final y co-ordinate of the swipe movement
Length	The number of data instances recorded during the swipe movement
Duration	The duration of the swipe movement in seconds

Table 2: Data Features of the training and test datasets

4. Analysis and Discussion

In this chapter, the results of the initial data analysis are discussed and compared with the results of several of the experiments covered in the literature review. The outcome of the training and testing of the various machine learning models are also presented and evaluated.

4.1. Significance Testing

Figure 14 to Figure 25 show the boxplots (without outliers) of the kinematic measurements which were calculated for each swipe gesture, and length. The left-hand boxplot represents the data from the TD group and the right-hand represents the ASD group. The TD group had higher mean (red dashed line) and median (solid black line) values for all measurements except gesture length and duration.

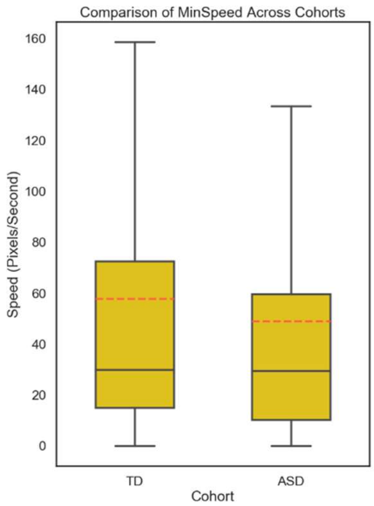


Figure 14: Boxplots of minimum speed for the two classes

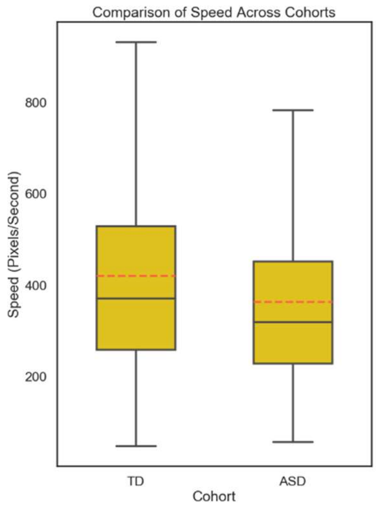


Figure 15: Boxplots of mean speed for the two classes

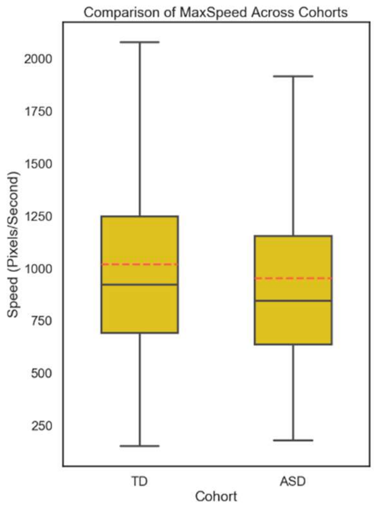


Figure 16: Boxplots of maximum speed for the two classes

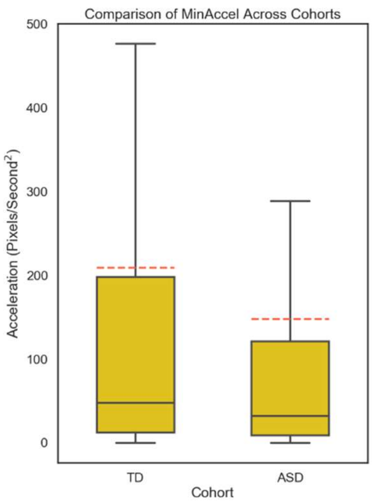


Figure 17: Boxplots of minimum acceleration for the two classes

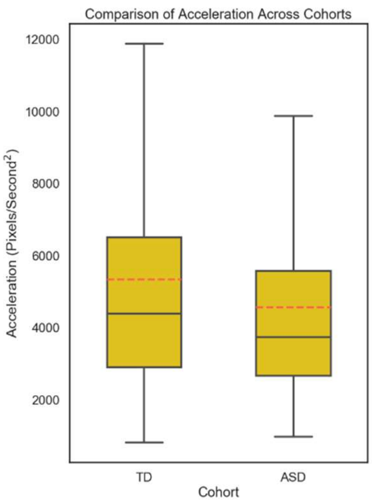


Figure 18: Boxplots of mean acceleration for the two classes

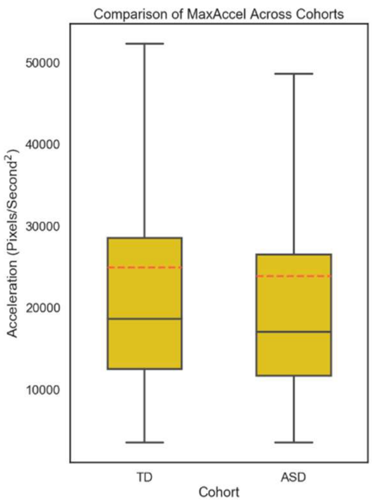


Figure 19: Boxplots of maximum acceleration for the two classes

The Mann-Whitney U test was used to check if these differences between the classes were significant for each measure. This test was chosen as the data is nonparametric and non-normally distributed (see appendix for histograms of data distributions). The null hypothesis, H_0, is that the differences are not significant and the two groups of data come from the same population. The p values from the Mann-Whitney U test for each measurement are shown in Table 3.

Every result was significant at the $\alpha = 0.5$ level, indicating the null hypothesis should be rejected, except for distance. As the gestures are goal-oriented and defined as starting and ending in specific regions it is unsurprising that there is not a significant difference in the distance of each swipe. Distance was removed as a feature from the dataset for modelling.

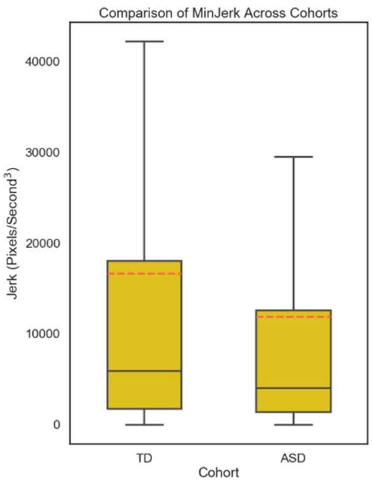


Figure 20: Boxplots of minimum jerk for the two classes

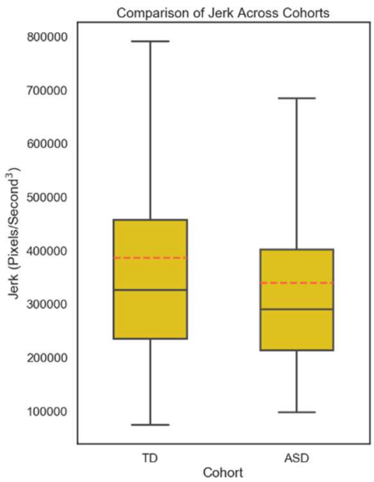


Figure 21: Boxplots of mean jerk for the two classes

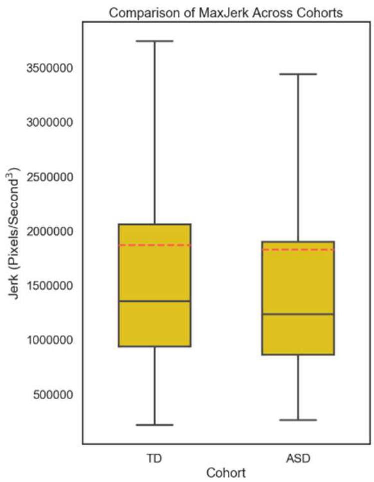


Figure 22: Boxplots of maximum jerk for the two classes

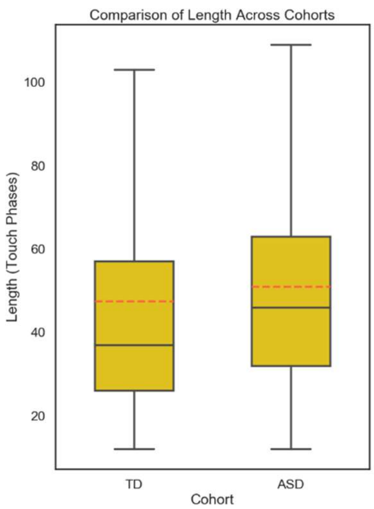


Figure 23: Boxplots of gesture for the two classes

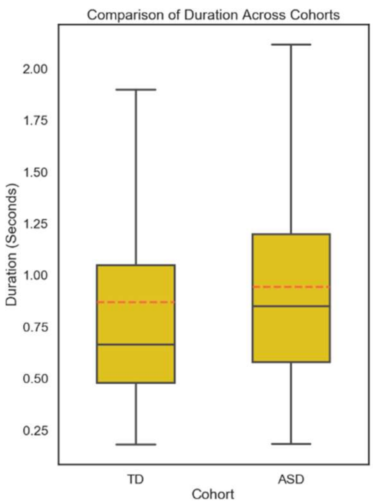


Figure 24: Boxplots of gesture duration for the two classes

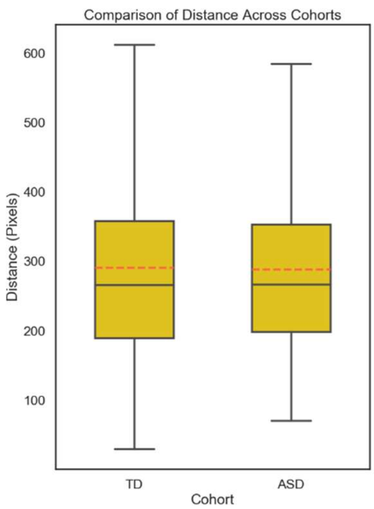


Figure 25: Boxplots of gesture distance for the two classes

These findings align with the work of Mari *et al.* (2003) and Glazebrook, Elliott and Lyons (2006) who found that the subjects with an autism diagnosis in their experiments had significantly higher maximum velocity than the control group, and acceleration in the case of Glazebrook et al, and significantly longer movement durations – which was also found by Forti *et al.* (2011).

Variable	p Value
Minimum Speed	3.05E-11 *
Mean Speed	1.09E-17 *
Maximum Speed	8.32E-08 *
Minimum Acceleration	1.13E-10 *
Mean Acceleration	2.71E-13 *
Maximum Acceleration	5.28E-05 *
Minimum Jerk	8.28E-10 *
Mean Jerk	1.38E-12 *
Maximum Jerk	7.38E-06 *
Length	5.72E-20 *
Distance	1.71E-01
Duration	8.07E-22 *

*Table 3: Results of the Mann Whitney U Test, * indicates significance*

4.2. Modelling Results

Choice of Evaluation Metrics

Classification accuracy, the number of correct predictions as a proportion of total predictions expressed as a percentage, is a common metric used to evaluate machine learning classifiers, however, it is not a good metric to use when classes are imbalanced, which they are in this case. Precision, Recall, and the F_1 score are much more informative in these circumstances.

In a binary classification task, each prediction falls into one of four categories – *true positives, true negatives, false positives* and *false negatives*. In this instance a *true positive* refers to a swipe gesture made by a child in the ASD group which is correctly predicted to belong to that group, and a *true negative* is a gesture made by a child in the control group which has been correctly labelled as such. A *false negative* would be a swipe gesture made by a child with autism being predicted as being made by a child without autism and a *false positive* would be the classification of a typically developing child's gesture as belonging to a child with ASD.

Precision is the number of gestures correctly labelled as coming from the ASD group as a proportion of all gestures labelled as belonging to that group - Equation 9, while recall is the proportion of correctly labelled ASD gestures as a proportion of the total number of gestures made by children in the ASD group - Equation 10. Values for both measures fall into the range of 0 to 1, with values closer to 1 being most desirable. The F_1 score is the harmonic mean of the precision and recall scores - Equation 11.

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

Equation 9

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

Equation 10

$$F_1 = \frac{precision\ \times recall}{precision + recall}$$

Equation 11

The aim of this project was to see if the Sharing game could be used in the screening or diagnosis of autism in young children. As such, it is more important to minimise false negatives than false positives. Typically developing children who are incorrectly identified by the game as possibly having autism can be "filtered out" at a later stage of the diagnostic process, however children with autism who are missed at this screening stage might lose their only chance for an early diagnosis, and thus lose out on the potential improvements to their prognosis that this brings. For this reason, recall was chosen as the metric used by the *GridSearchCV* function when assessing which hyperparameters produce the best models for this task.

Support Vector Machine

The first model to be built was a Support Vector Machine with a Gaussian RBF kernel, and the class weights hyperparameter set that both classes held equal weight - SVM RBF_1. The grid search function was used to find the best value for the C and γ hyperparameters, these were found to be C = 10 and γ = 1.

The classification report and confusion matrix for this model are shown in Table 4 and Table 5 respectively. The recall for the TD class is very good, but for the ASD class it is extremely low. The overall F_1 score of 0.62 means there is room for improvement. From the confusion matrix, Table 5, it can be seen that the model is predicting the vast majority of data instances as belonging to the TD class.

	Precision	Recall	F_1 Score	Support	
TD	0.70	0.83	0.76	317	
ASD	0.45	0.28	0.34	159	
Average	0.61	0.64	0.62	Total	476

Table 4: Classification Report for SVM RBF_1 Model

	Predicted as TD	Predicted as ASD
Actually TD	263	54
Actually ASD	115	44

Table 5: Confusion Matrix for SVM RBF_1 Model

In an attempt to counteract this, a second SVM with a Gaussian RBF kernel was built - SVM RBF_2, this time with the class weights adjusted to give the ASD class twice the weight of the TD class, as the test set contains two TD gestures for every ASD labelled gesture.

The grid search function found the best parameters for this model to be C = 1 and γ to take the "auto" value defined by the Scikit-Learn function as the reciprocal of the number of features in the training set, in this instance $\frac{1}{15} \approx 0.067$.

The result of making this alteration can be seen in Table 6 and Table 7. The recall for the ASD class is much higher now, at 0.91 almost all ASD gestures are being correctly classified, however this comes at the expense of the recall for the TD class which is now only 0.18. The model hasn't got better at differentiating between classes, it is just predicting almost everything as belonging to the ASD class instead of the TD class.

	Precision	Recall	F_1 Score	Support	
TD	0.79	0.18	0.29	317	
ASD	0.36	0.91	0.51	159	
Average	0.64	0.42	0.36	Total	476

Table 6: Classification Report for SVM RBF_2

	Predicted as TD	Predicted as ASD
Actually TD	56	261
Actually ASD	15	144

Table 7: Confusion Matrix for SVM RBF_2

Another Support Vector Machine was built, this time with a linear kernel. The classes were given equal weighting and the grid search function determined the best value of C to be 1. The results from this model are shown in Table 8 and Table 9.

Overall this model is poor. It has higher recall for the ASD class than the equally weighted Gaussian RBF SVM, however it still misclassifies nearly half the data. It is not really an improvement on the previous models.

	Precision	Recall	F_1 Score	Support	
TD	0.73	0.42	0.53	317	
ASD	0.37	0.69	0.48	159	
Average	0.61	0.51	0.51	Total	476

Table 8: Classification Report for Linear SVM

	Predicted as TD	Predicted as ASD
Actually TD	132	185
Actually ASD	50	109

Table 9: Confusion Matrix for Linear SVM

Table 10 shows the classification report for a Support Vector machine built using the polynomial kernel, and the corresponding confusion matrix is shown in Table 11. A grid search found the parameters for this type of model which gave the best recall score to be C = 1 and the degree of the polynomial to be 10.

This model has fantastic recall for the ASD class, however once again this is at the expense of the precision. The model is simply predicting the majority of data instances as belonging to the ASD class and is not performing particularly well overall.

	Precision	Recall	F_1 Score	Support	
TD	0.77	0.15	0.25	317	
ASD	0.35	0.91	0.51	159	
Average	0.63	0.41	0.34	Total	476

Table 10: Classification Report for Polynomial SVM

	Predicted as TD	Predicted as ASD
Actually TD	48	269
Actually ASD	14	145

Table 11: Confusion Matrix for Polynomial SVM

Decision Tree

The next type of model which was built was a Decision Tree. Using a grid search method, the optimal parameters for the Tree were found to be a maximum depth of 20, a minimum of two samples for a node to be split, and with the impurity measure set to *gini* rather than *entropy*. All other parameters were set to their default values within the Scikit-Learn *DecisionTreeModel* function.

The classification report of this model can be seen in Table 12 and the confusion matrix in Table 13. The F_1 score for the model, 0.60 is quite low – however it is better than all but one of the Support Vector Machine models which were built. The recall for ASD labelled gestures is only 0.47, meaning 53%, over half, of gestures made by children with autism were incorrectly classified as belonging to the control group.

	Precision	Recall	F_1 Score	Support	
TD	0.71	0.65	0.68	317	
ASD	0.40	0.47	0.43	159	
Average	0.61	0.59	0.60	Total	476

Table 12: Classification Report for the Decision Tree Model

	Predicted as TD	Predicted as ASD
Actually TD	206	111
Actually ASD	84	75

Table 13: Confusion Matrix for Decision Tree Model

Bagging

In an effort to improve upon these results, a predictor was built using the Decision Tree model and the Bagging algorithm. The grid search method was used to find the value for the number of estimators (i.e. trees) which gave the best recall values. This was found to be 250. The classification report and confusion matrix for this model can be seen in Table 14 and Table 15 respectively.

As expected, there has been an improvement in performance, with an F_1 of 0.66 compared to 0.60 for the Decision Tree model, and an improvement on precision for both classes. However, while the average recall values for the model have improved, the recall value for the ASD class is actually lower.

	Precision	Recall	F_1 Score	Support	
TD	0.74	0.79	0.76	317	
ASD	0.51	0.43	0.47	159	
Average	0.66	0.67	0.66	Total	476

Table 14: Classification Report for Bagging Model

	Predicted as TD	Predicted as ASD
Actually TD	251	66
Actually ASD	90	69

Table 15: Confusion Matrix for Bagging Model

Random Forest

While using a similar technique to Bagging, a Random Forest model is expected to perform better due to the introduction of a random element – as explained in the methodology chapter. Once again, the grid search function was used to find the best parameters for the Random Forest model, which would maximise the recall score. The model which performed best had entropy for its impurity measure, a maximum depth of 15, a minimum of 2 samples for a node to be split, and 500 estimators.

Table 16 and Table 17 show the classification report and confusion matrix for this optimal model. There is once again an improvement on the previous model, this method has produced the best model – with an overall F_1 score of 0.68. The recall for members of the ASD class is 0.53, slightly more than half. 47% of swipes made by children with autism are being misclassified.

	Precision	Recall	F_1 Score	Support	
TD	0.76	0.75	0.75	317	
ASD	0.51	0.53	0.52	159	
Average	0.68	0.67	0.68	Total	476

Table 16: Classification Report for the Random Forest Model

	Predicted as TD	Predicted as ASD
Actually TD	237	80
Actually ASD	75	84

Table 17: Confusion Matrix for the Random Forest Model

When making predictions, the Random Forest classifier calculates a probability for each data instance in the test set belonging to each class. The probability threshold for predicting a class is automatically set at 0.5. It is possible to trade-off between precision and recall within a model by adjusting this threshold, for example the number of ASD instances being misclassified can be reduced at the cost of having more TD instances being misclassified. As mentioned previously, it is better to include some children without autism at the early screening stage, who can be excluded at a later stage, than to miss some children who do have autism who might then lose out on important early interventions.

Some investigation was conducted with the Random Forest model to see if moving this threshold slightly could improve the results, however it was found that to ensure zero ASD gestures were misclassified as TD gestures the probability threshold would have to be reduced to 0.16, at which point 96.5% of TD gestures were being misclassified as belonging to the ASD class, and only 3.5% of TD gestures were being correctly classified. In the real world this would be extremely ineffective as a screening measure as very few children would actually be excluded. Even if a recall value of 0.9 for the ASD class was chosen as acceptable for the model this would require the threshold to be dropped to 0.28 and still 82.6% of the TD class would be misclassified as belonging to the ASD class. The code to perform this threshold analysis was adapted from (Arvai, 2018).

A Precision-Recall curve for the Random Forest model is shown in Figure 26. It can be seen that 100% precision can only occur with this model at the expense of recall being extremely low, around 5%. This high precision drops off very sharply as recall increases, with it remaining around 40-50% as recall increases to 100%.

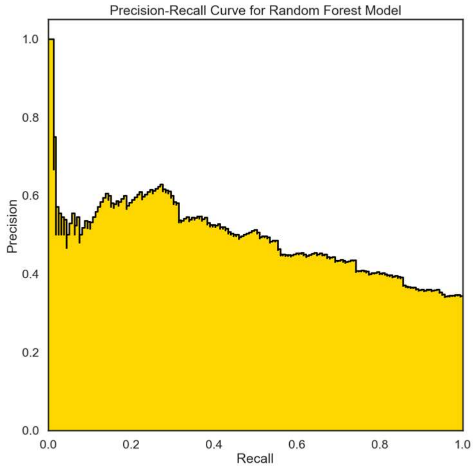


Figure 26: Precision-Recall Curve for the Random Forest Model

Figure 27 shows the importance that the model put on each of the features in the data set – all importance values sum to one. No feature stands out as being significantly more important than any other, all features have similar importance values. Duration was the most important feature to the model, this was also the measurement that had the most significant p value when tested with the Mann-Whitney U test - Table 3. The next most important features were the final co-ordinates of the swipe gesture – perhaps one of the groups of children showed higher accuracy of placing the food items on the plates in the game. There is scope to investigate this further in future work.

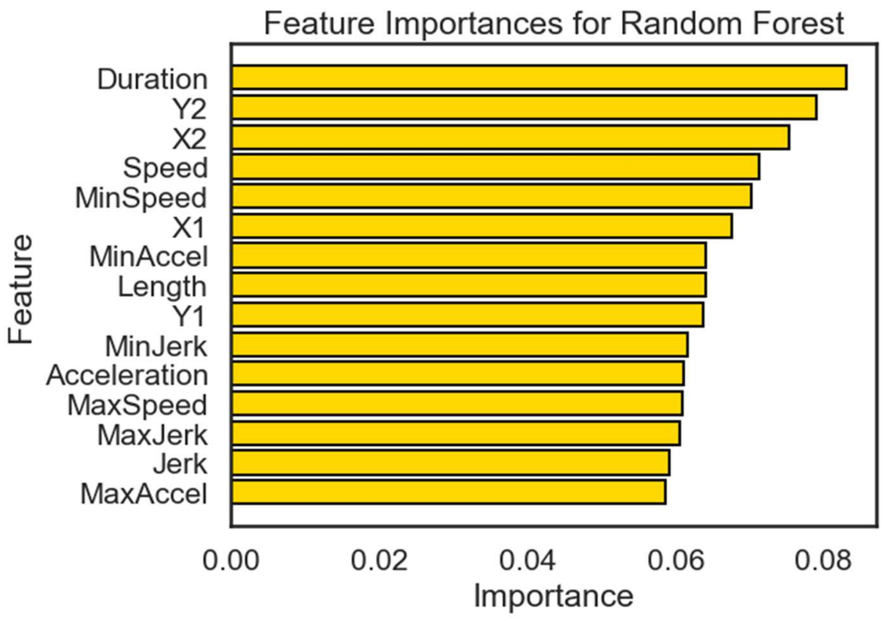


Figure 27: Feature Importance for the Random Forest Model

5. Conclusions

The aim of this project was to explore whether iPad games could be used to aid in the early diagnosis of autism in young children. This was reliant on there being significant differences in how children with autism and their typically developing counterparts interact with the game.

The task was to train machine learning classifiers on this data, in the hope that the models could successfully determine which gestures were made by which group of children.

The initial analysis of the data, and the subsequent calculations of various kinematic measurements for each completed swipe gesture, found that there were significant differences between the children with an autism spectrum disorder diagnosis and the children in the control group for all measurements except swipe distance. The most significant difference was found with the gesture duration measures, a finding which aligns with a lot of the existing literature in this area, lending further proof to the theory that children with autism have motor issues which cause them to take longer to execute goal-oriented tasks.

These significantly different kinematic measures were used as data features to train seven machine learning models, the best model – the Random Forest, achieved an F_1 score of 0.68, with an average recall of 0.67. The recall for the ASD class was 53% - something that can be built on in future work.

The results from the machine learning task are promising and finding that there were significant differences in the kinematics of the swipe gestures is valuable research which aligns with the literature and lends credence to the idea that a Smart Game diagnosis tool is possible.

5.1. Future Work

Training and testing the model on more data is an obvious first step in extending the work carried out in this project. Other data features could also be investigated, such as utilising some of the sensor data from the iPad - ensuring this data is relevant to the subjects' movement and not external or accidental factors such as a table getting knocked. An example of a variable that could be of interest is the force being exerted on the screen by a child during a movement.

This project has shown that there are significant differences in the gesture kinematics between typically developing children and their peers with autism. This could be investigated further, and perhaps more accurate models could be built, if a new game were developed for the collection of movement data. The goal-oriented swipes created by players of the Sharing game were short, and often restricted - a game which allowed for longer or more variable swipes could make the differences between the children with and without autism more pronounced.

This researched focussed solely on classifying individual goal-oriented gestures, assuming that they are independent of all other gestures, however they are not. The children in the study made on average dozens of gestures each. While perhaps it is not currently possible to correctly classify swipes as either

TD or ASD with 100% accuracy, the classification of children could be possible by looking at the proportions of their swipes being classified into each group. There might exist a percentage threshold of swipes being classified as ASD which can be used to separate the children into the two classes.

Researching this theory would be possible with data from more subjects, as it would be necessary to test on, and classify, the entire set of gestures created by a child, and repeat this for many children to look for a pattern or threshold.

Overall, this project has produced promising results which not only build on, and validate, the existing research in this area but provide a solid foundation for further research to be carried out.

6. References

Adachi, M. *et al.* (2018) 'Adaptation of the Autism Spectrum Screening Questionnaire (ASSQ) to preschool children', *PLOS ONE*. Public Library of Science, 13(7), pp. 1–19. doi: 10.1371/journal.pone.0199590.

Ahearne, C. *et al.* (2016) 'Touch-screen technology usage in toddlers', *Archives of Disease in Childhood*. doi: 10.1136/archdischild-2015-309278.

Akshoomoff, N., Corsello, C. and Schmidt, H. (2006) 'The Role of the Autism Diagnostic Observation Schedule in the Assessment of Autism Spectrum Disorders in School and Community Settings.', *The California school psychologist : CASP / California Association of School Psychologists*. doi: 10.1007/BF03341111.

American Psychiatric Association (2013) *Diagnostic and statistical manual of mental disorders (5th ed.)*, *American Journal of Psychiatry*. doi: 10.1176/appi.books.9780890425596.744053.

Anzulewicz, A., Sobota, K. and Delafield-Butt, J. T. (2016) 'Toward the Autism Motor Signature: Gesture patterns during smart tablet gameplay identify children with autism', *Scientific Reports*. doi: 10.1038/srep31107.

Arvai, K. (2018) *Fine tuning a classifier in scikit-learn*, *Towards Data Science*. Available at: https://towardsdatascience.com/fine-tuning-a-classifier-in-scikit-learn-66e048c21e65 (Accessed: 27 August 2018).

Baird, G. *et al.* (2006) 'Prevalence of disorders of the autism spectrum in a population cohort of children in South Thames: the Special Needs and Autism Project (SNAP)', *Lancet*. doi: 10.1016/S0140-6736(06)69041-7.

Baron-Cohen, S. *et al.* (2009) 'Prevalence of autism-spectrum conditions: UK school-based population study', *British Journal of Psychiatry*. doi: 10.1192/bjp.bp.108.059345.

Bevan, C. and Fraser, D. S. (2016) 'Different strokes for different folks? Revealing the physical characteristics of smartphone users from their swipe gestures', *International Journal of Human Computer Studies*. doi: 10.1016/j.ijhcs.2016.01.001.

Bhat, A. N., Landa, R. J. and Galloway, J. C. (2011) 'Current Perspectives on Motor Functioning in Infants, Children, and Adults With Autism Spectrum Disorders', *Physical Therapy*. doi: 10.2522/ptj.20100294.

Bone, D. *et al.* (2015) 'Applying Machine Learning to Facilitate Autism Diagnostics: Pitfalls and Promises', *Journal of Autism and Developmental Disorders*. doi: 10.1007/s10803-014-2268-6.

Bone, D. *et al.* (2016) 'Use of machine learning to improve autism screening and diagnostic instruments: effectiveness, efficiency, and multi-instrument fusion', *Journal of Child Psychology and Psychiatry and Allied Disciplines*. doi: 10.1111/jcpp.12559.

Bosl, W. *et al.* (2011) 'EEG complexity as a biomarker for autism spectrum disorder risk', *BMC Medicine*.

doi: 10.1186/1741-7015-9-18.

Brett, D. *et al.* (2016) ‘Factors Affecting Age at ASD Diagnosis in UK: No Evidence that Diagnosis Age has Decreased Between 2004 and 2014’, *Journal of Autism and Developmental Disorders*. doi: 10.1007/s10803-016-2716-6.

Brian, J. *et al.* (2008) ‘Clinical assessment of autism in high-risk 18-month-olds’, *Autism*. doi: 10.1177/1362361308094500.

Brugha, T. S. *et al.* (2011) ‘Epidemiology of autism spectrum disorders in adults in the community in England’, *Archives of General Psychiatry*. doi: 10.1001/archgenpsychiatry.2011.38.

Bryson, S. E. *et al.* (2008) ‘The autism observation scale for infants: Scale development and reliability data’, *Journal of Autism and Developmental Disorders*. doi: 10.1007/s10803-007-0440-y.

Cook, J. (2016) ‘From movement kinematics to social cognition: The case of autism’, *Philosophical Transactions of the Royal Society B: Biological Sciences*. doi: 10.1098/rstb.2015.0372.

Cook, J. L., Blakemore, S. J. and Press, C. (2013) ‘Atypical basic movement kinematics in autism spectrum conditions’, *Brain*. doi: 10.1093/brain/awt208.

Crane, L. *et al.* (2016) ‘Experiences of autism diagnosis: A survey of over 1000 parents in the United Kingdom’, *Autism*. doi: 10.1177/1362361315573636.

Crippa, A. *et al.* (2013) ‘Eye-hand coordination in children with high functioning autism and Asperger’s disorder using a gap-overlap paradigm’, *Journal of Autism and Developmental Disorders*. doi: 10.1007/s10803-012-1623-8.

Crippa, A. *et al.* (2015) ‘Use of Machine Learning to Identify Children with Autism and Their Motor Abnormalities’, *Journal of Autism and Developmental Disorders*. doi: 10.1007/s10803-015-2379-8.

Deo, R. C. (2015) ‘Machine learning in medicine’, *Circulation*. doi: 10.1161/CIRCULATIONAHA.115.001593.

Deshpande, G. *et al.* (2013) ‘Identification of neural connectivity signatures of autism using machine learning’, *Frontiers in Human Neuroscience*. doi: 10.3389/fnhum.2013.00670.

Ehlers, S., Gillberg, C. and Wing, L. (1999) ‘A Screening Questionnaire for Asperger Syndrome and Other High-Functioning Autism Spectrum Disorders in School Age Children’, *Journal of Autism and Developmental Disorders*. doi: 10.1023/A:1023040610384.

Eikeseth, S. *et al.* (2012) ‘Outcome for children with autism receiving early and intensive behavioral intervention in mainstream preschool and kindergarten settings’, *Research in Autism Spectrum Disorders*. doi: 10.1016/j.rasd.2011.09.002.

Forti, S. *et al.* (2011) ‘Motor planning and control in autism. A kinematic analysis of preschool children’, *Research in Autism Spectrum Disorders*. doi: 10.1016/j.rasd.2010.09.013.

Foster, K. R., Koprowski, R. and Skufca, J. D. (2014) 'Machine learning, medical diagnosis, and biomedical engineering research - commentary', *BioMedical Engineering Online*. doi: 10.1186/1475-925X-13-94.

Fountain, C., King, M. D. and Bearman, P. S. (2011) 'Age of diagnosis for autism: Individual and community factors across 10 birth cohorts', *Journal of Epidemiology and Community Health*. doi: 10.1136/jech.2009.104588.

Fournier, K. A. *et al.* (2010) 'Motor coordination in autism spectrum disorders: A synthesis and meta-analysis', *Journal of Autism and Developmental Disorders*. doi: 10.1007/s10803-010-0981-3.

Glazebrook, C. M., Elliott, D. and Lyons, J. (2006) 'A kinematic analysis of how young adults with and without autism plan and control goal-directed movements.', *Motor control*. doi: 10.1123/mcj.10.3.244.

Goin-Kochel, R. P., Mackintosh, V. H. and Myers, B. J. (2006) 'How many doctors does it take to make an autism spectrum diagnosis?', *Autism*. doi: 10.1177/1362361306066601.

Gowen, E. and Hamilton, A. (2013) 'Motor abilities in autism: A review using a computational context', *Journal of Autism and Developmental Disorders*. doi: 10.1007/s10803-012-1574-0.

Green, D. *et al.* (2009) 'Impairment in movement skills of children with autistic spectrum disorders', *Developmental Medicine and Child Neurology*. doi: 10.1111/j.1469-8749.2008.03242.x.

Hardy, S. *et al.* (2015) 'Can Screening with the Ages and Stages Questionnaire Detect Autism?', *Journal of Developmental and Behavioral Pediatrics*. doi: 10.1097/DBP.0000000000000201.

Ingalhalikar, M. *et al.* (2011) 'Diffusion based abnormality markers of pathology: Toward learned diagnostic prediction of ASD', *NeuroImage*. doi: 10.1016/j.neuroimage.2011.05.023.

Ben Itzchak, E. and Zachor, D. A. (2011) 'Who benefits from early intervention in autism spectrum disorders?', *Research in Autism Spectrum Disorders*. doi: 10.1016/j.rasd.2010.04.018.

Johnson, B. P. *et al.* (2013) 'Understanding macrographia in children with autism spectrum disorders', *Research in Developmental Disabilities*. doi: 10.1016/j.ridd.2013.06.003.

Kabali, H. K. *et al.* (2015) 'Exposure and Use of Mobile Media Devices by Young Children', *PEDIATRICS*. doi: 10.1542/peds.2015-2151.

Kim, Y. S. *et al.* (2011) 'Prevalence of Autism Spectrum Disorders in a Total Population Sample', *American Journal of Psychiatry*. doi: 10.1176/appi.ajp.2011.10101532.

Kosmicki, J. A. *et al.* (2015) 'Searching for a minimal set of behaviors for autism detection through feature selection-based machine learning', *Translational Psychiatry*. doi: 10.1038/tp.2015.7.

Lai, M., Lombardo, M. V and Baron-Cohen, S. (2014) 'Autism.', *Lancet*. doi: 10.1016/S0140-6736(13)61539-1.

Lange, N. *et al.* (2010) ‘Atypical diffusion tensor hemispheric asymmetry in autism’, *Autism Research*. doi: 10.1002/aur.162.

Liu, W., Li, M. and Yi, L. (2016) ‘Identifying children with autism spectrum disorder based on their face processing abnormality: A machine learning framework’, *Autism Research*. doi: 10.1002/aur.1615.

Lord, C. *et al.* (2000) ‘Autism Diagnostic Observation Schedule (ADOS)’, *Journal of Autism and Developmental Disorders*. doi: 10.1007/BF02211841.

Lord, C., Rutter, M. and Le Couteur, A. (1994) ‘Autism Diagnostic Interview-Revised: A revised version of a diagnostic interview for caregivers of individuals with possible pervasive developmental disorders’, *Journal of Autism and Developmental Disorders*. doi: 10.1007/BF02172145.

Lord, C. and Volkmar, F. R. (2007) ‘Diagnosis and definition of autism and other pervasive developmental disorders’, in Volkmar, F. R. (ed.) *Autism and Pervasive Developmental Disorders*. Cambridge: Cambridge University Press, pp. 1–32. doi: 10.1017/CBO9780511544446.002.

MacDonald, R. *et al.* (2014) ‘Assessing progress and outcome of early intensive behavioral intervention for toddlers with autism’, *Research in Developmental Disabilities*. doi: 10.1016/j.ridd.2014.08.036.

Mahmoudi-Nejad, A., Moradi, H. and Pouretemad, H. R. (2017) ‘The differences between children with autism and typically developed children in using a hand-eye-coordination video game’, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. doi: 10.1007/978-3-319-67585-5_27.

Maity, N. G., Das, S. and Ieee (2017) ‘Machine Learning for Improved Diagnosis and Prognosis in Healthcare’, in *2017 Ieee Aerospace Conference*.

Mari, M. *et al.* (2003) ‘The reach-to-grasp movement in children with autism spectrum disorder.’, *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*. doi: 10.1098/rstb.2002.1205.

Marvin, A. R. *et al.* (2017) ‘Analysis of Social Communication Questionnaire (SCQ) Screening for Children Less Than Age 4’, *Current Developmental Disorders Reports*. doi: 10.1007/s40474-017-0122-1.

Mattila, M. L. *et al.* (2011) ‘Autism spectrum disorders according to DSM-IV-TR and comparison with DSM-5 draft criteria: An epidemiological study’, *Journal of the American Academy of Child and Adolescent Psychiatry*. doi: 10.1016/j.jaac.2011.04.001.

Mattila, M. L. *et al.* (2012) ‘Validation of the Finnish Autism Spectrum Screening Questionnaire (ASSQ) for clinical settings and total population screening’, *Journal of Autism and Developmental Disorders*. doi: 10.1007/s10803-012-1464-5.

Miguel-Hurtado, O. *et al.* (2016) ‘Predicting sex as a soft-biometrics from device interaction swipe gestures’, *Pattern Recognition Letters*. doi: 10.1016/j.patrec.2016.04.024.

Motlagh, S. H. R. E., Moradi, H. and Pouretemad, H. (2013) 'Using general sound descriptors for early autism detection', in *2013 9th Asian Control Conference, ASCC 2013*. doi: 10.1109/ASCC.2013.6606386.

Oller, D. K. *et al.* (2010) 'Automated vocal analysis of naturalistic recordings from children with autism, language delay, and typical development', *Proceedings of the National Academy of Sciences*. doi: 10.1073/pnas.1003882107.

Palmer, E. *et al.* (2011) 'Autism spectrum disorder diagnostic assessments: Improvements since publication of the National Autism Plan for Children', *Archives of Disease in Childhood*. doi: 10.1136/adc.2009.172825.

Robins, D. L. *et al.* (2001) 'The Modified Checklist for Autism in Toddlers: An Initial Study Investigating the Early Detection of Autism and Pervasive Developmental Disorders', *Journal of Autism and Developmental Disorders*. doi: 10.1023/A:1010738829569.

Rogers, S. J. and Vismara, L. A. (2008) 'Evidence-based comprehensive treatments for early autism', *Journal of Clinical Child and Adolescent Psychology*. doi: 10.1080/15374410701817808.

Rubenstein, E. *et al.* (2017) 'The Broader Autism Phenotype in Mothers is Associated with Increased Discordance Between Maternal-Reported and Clinician-Observed Instruments that Measure Child Autism Spectrum Disorder', *Journal of Autism and Developmental Disorders*. doi: 10.1007/s10803-017-3248-4.

Rudra, A. *et al.* (2014) 'G308 A school-based cohort study of autistic symptoms in 3-8 year olds in India based on parent and teacher response to internationally standardised screening questionnaires', *Archives of Disease in Childhood*. doi: http://dx.doi.org/10.1136/archdischild-2014-306237.291.

Russell, G. *et al.* (2014) 'Prevalence of parent-reported ASD and ADHD in the UK: Findings from the millennium cohort study', *Journal of Autism and Developmental Disorders*. doi: 10.1007/s10803-013-1849-0.

Rutter Bailey, A., & Lord, C., M. (2003) 'Social Communication Questionnaire', *Los Angeles, CA*.

Rutter, M., Bailey, A. and Lord, C. (2003) 'Social Communication Questionnaire.(SCQ): Manual. West', *Psychol. Serv, Los Angeles*.

Sajda, P. (2006) 'Machine learning for detection and diagnosis of disease', *Annual Review of Biomedical Engineering*. doi: 10.1146/annurev.bioeng.8.061505.095802.

Sato, J. R. *et al.* (2013) 'Inter-regional cortical thickness correlations are associated with autistic symptoms: A machine-learning approach', *Journal of Psychiatric Research*. doi: 10.1016/j.jpsychires.2012.11.017.

Shattuck, P. T. *et al.* (2009) 'Timing of identification among children with an autism spectrum disorder: Findings from a population-based surveillance study', *Journal of the American Academy of Child and Adolescent Psychiatry*. doi: 10.1097/CHI.0b013e31819b3848.

Sipes, M. and Matson, J. L. (2014) 'Measures Used to Screen and Diagnose ASD in Young Children', in Tarbox, J. et al. (eds) *Handbook of Early Intervention for Autism Spectrum Disorders: Research, Policy,* *and Practice*. New York, NY: Springer New York, pp. 87–103. doi: 10.1007/978-1-4939-0401-3_5.

Stahl, D. *et al.* (2012) 'Novel machine learning methods for ERP analysis: A validation from research on infants at risk for autism', *Developmental Neuropsychology*. doi: 10.1080/87565641.2011.650808.

Trevarthen, C. and Delafield-Butt, J. T. (2013) 'Autism as a developmental disorder in intentional movement and affective engagement', *Frontiers in Integrative Neuroscience*. doi: 10.3389/fnint.2013.00049.

Twomey, D. M. *et al.* (2018) 'Feasibility of using touch screen technology for early cognitive assessment in children', *Archives of Disease in Childhood*. BMJ Publishing Group Ltd. doi: 10.1136/archdischild-2017-314010.

Wall, D. P. *et al.* (2012) 'Use of artificial intelligence to shorten the behavioral diagnosis of autism', *PLoS ONE*. doi: 10.1371/journal.pone.0043855.

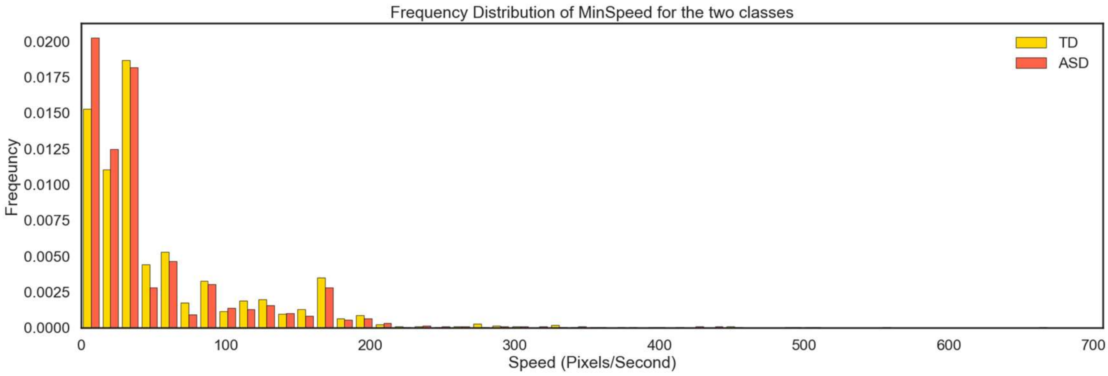
Wall, D. P. *et al.* (2012) 'Use of machine learning to shorten observation-based screening and diagnosis of autism', *Translational Psychiatry*. doi: 10.1038/tp.2012.10.

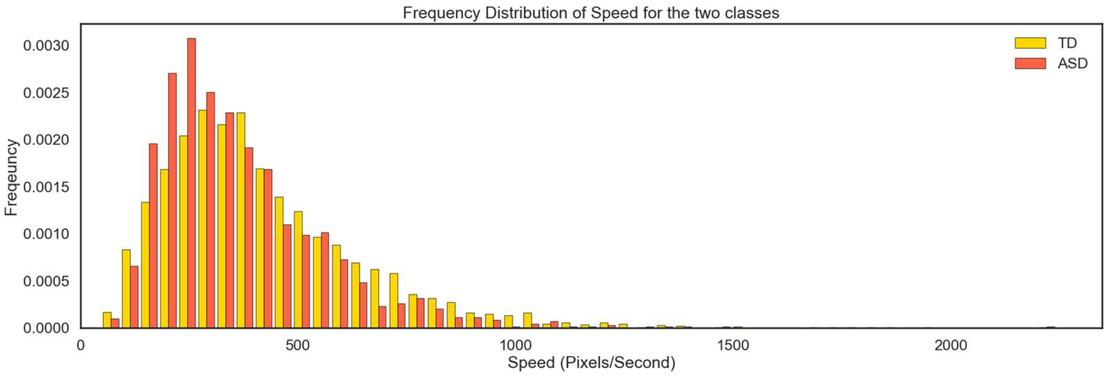
Xu, D. *et al.* (2009) 'Child vocalization composition as discriminant information for automatic autism detection', in *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine, EMBC 2009*. doi: 10.1109/IEMBS.2009.5334846.

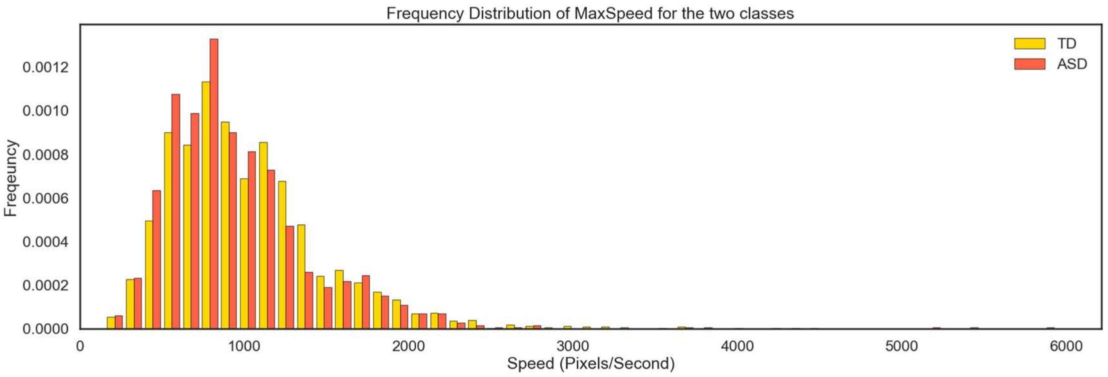
Zhou, Y., Yu, F. and Duong, T. (2014) 'Multiparametric MRI characterization and prediction in autism spectrum disorder using graph theory and machine learning', *PLoS ONE*. doi: 10.1371/journal.pone.0090405.

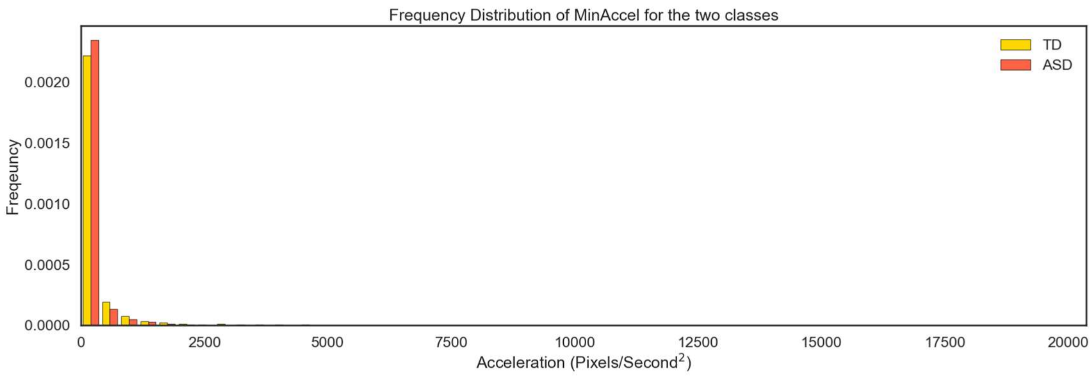
7. Appendices

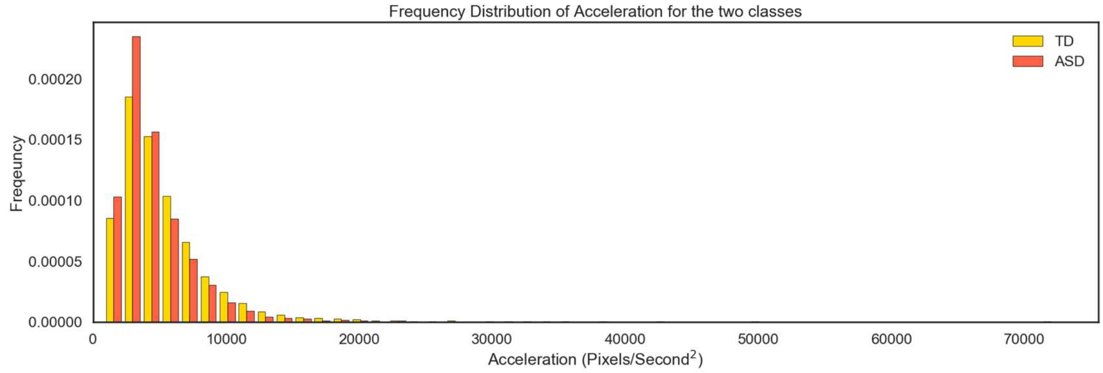
Frequency Distributions of Kinematic Measurements

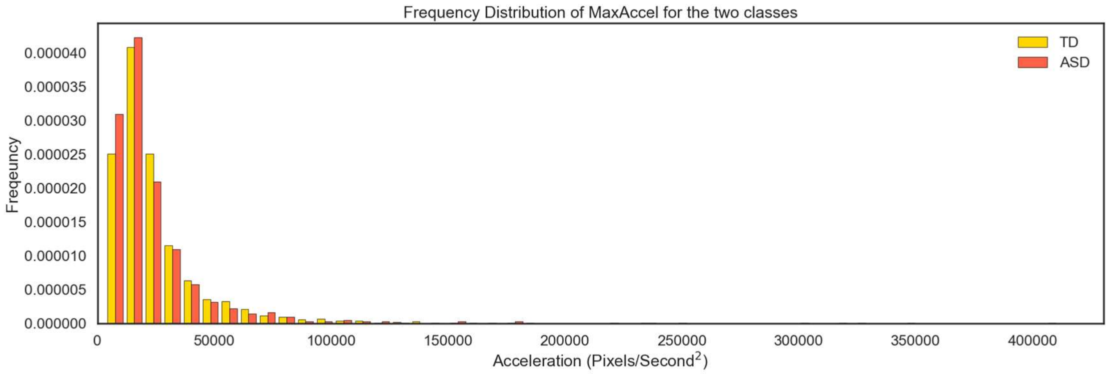


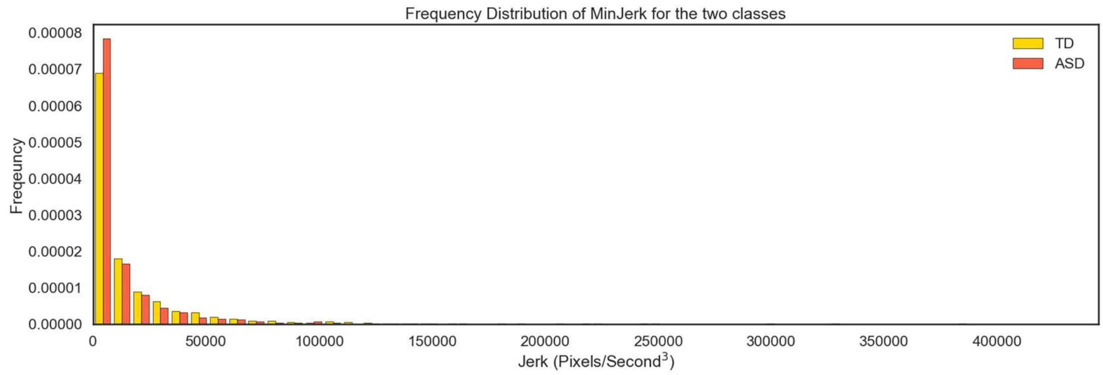


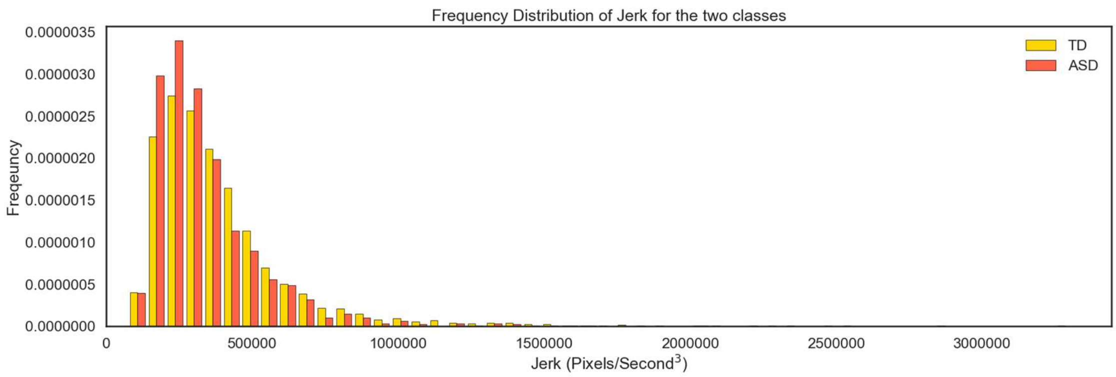


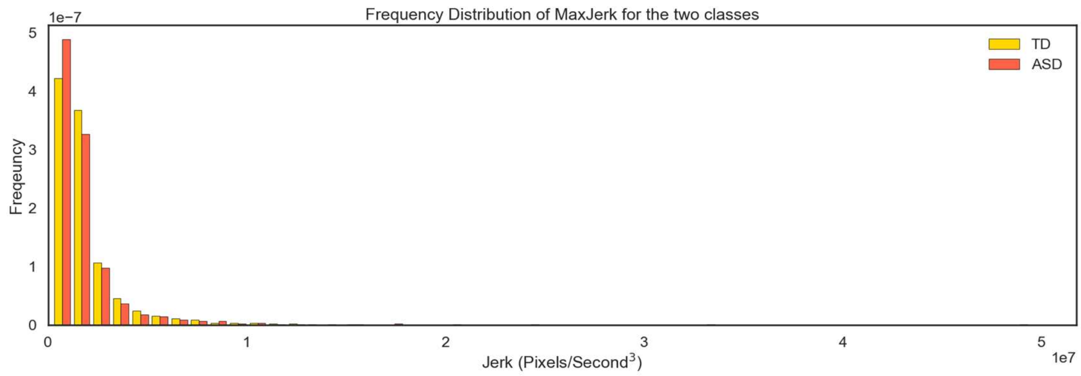


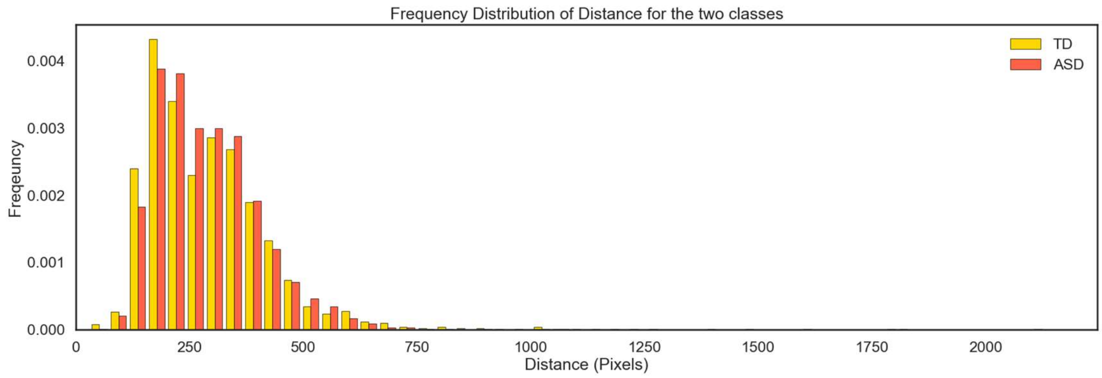

Frequency Distribution of Acceleration for the two classes
TD
ASD
Freqeuncy
0.00020
0.00015
0.00010
0.00005
0.00000
0
10000
20000
30000
40000
50000
60000
70000
Acceleration (Pixels/Second^2)

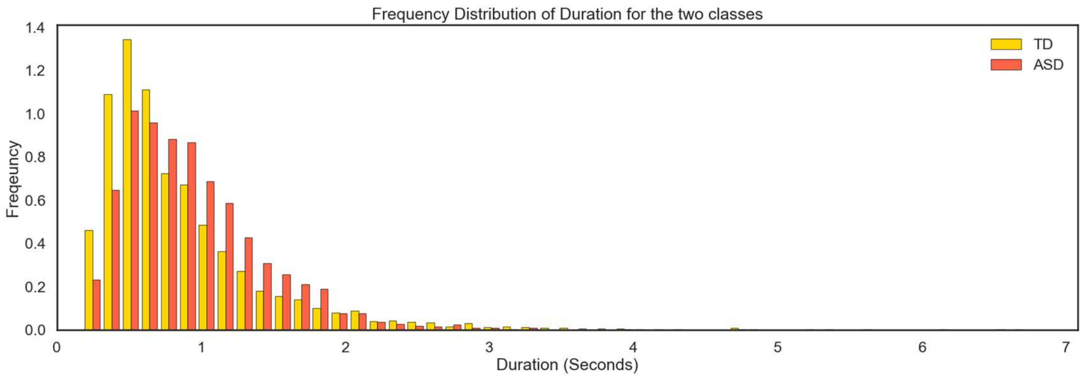

Frequency Distribution of MaxAccel for the two classes
TD
ASD
Freqeuncy
0.000040
0.000035
0.000030
0.000025
0.000020
0.000015
0.000010
0.000005
0.000000
0
50000
100000
150000
200000
250000
300000
350000
400000
Acceleration (Pixels/Second^2)

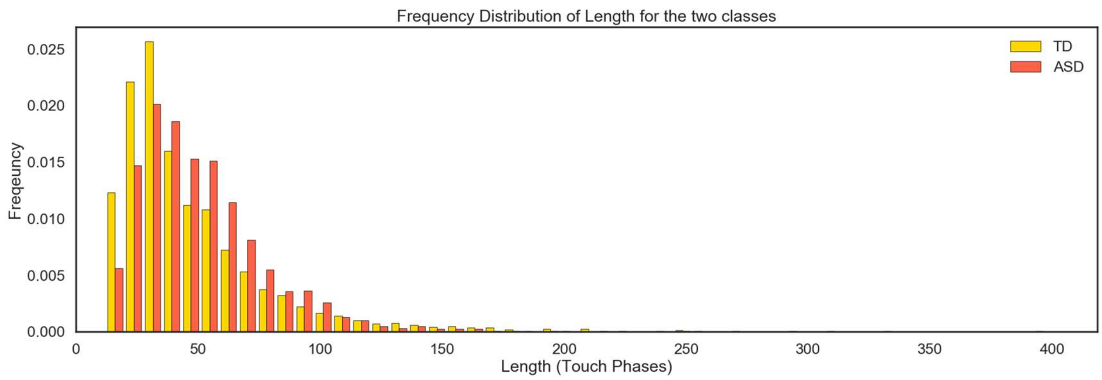

Frequency Distribution of MinJerk for the two classes
TD
ASD
Freqeuncy
0.00008
0.00007
0.00006
0.00005
0.00004
0.00003
0.00002
0.00001
0.00000
0
50000
100000
150000
200000
250000
300000
350000
400000
Jerk (Pixels/Second^3)


Frequency Distribution of Jerk for the two classes
TD
ASD
Freqeuncy
0.0000035
0.0000030
0.0000025
0.0000020
0.0000015
0.0000010
0.0000005
0.0000000
0
500000
1000000
1500000
2000000
2500000
3000000
Jerk (Pixels/Second^3)


Frequency Distribution of MaxJerk for the two classes
1e−7
Freqeuncy
Jerk (Pixels/Second^3)
1e7
TD
ASD


Frequency Distribution of Distance for the two classes
Freqeuncy
Distance (Pixels)
TD
ASD


Frequency Distribution of Duration for the two classes
Freqeuncy
Duration (Seconds)
TD
ASD


Frequency Distribution of Length for the two classes
Freqeuncy
Length (Touch Phases)
TD
ASD

Python Code

```
# -*- coding: utf-8 -*-
"""
Created on Fri Jul  6 12:31:50 2018

@author: poppy
"""
import seaborn as sns; sns.set()
import numpy as np
import math
import pandas as pd
import glob, os
import matplotlib.pyplot as plt
from collections import Counter

import scipy
from scipy import stats
from scipy import sparse
from scipy.stats import mannwhitneyu

import imblearn.over_sampling
from imblearn.over_sampling import RandomOverSampler

import sklearn as sklearn
from sklearn import metrics
from sklearn import datasets
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_recall_curve
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV

#PICKLES######################################################################
SwipeData = pd.read_pickle('SwipeData.pkl')
touch2 = pd.read_pickle('touch2.pkl')
#############################################################################

#read in ASD and TD data #################################################
asd = pd.concat(map(pd.read_csv, glob.glob(os.path.join('',
"D:\\study0_reorganized data_SL\\subject*_ASD_sharing_touch.csv"))))
td = pd.concat(map(pd.read_csv, glob.glob(os.path.join('', "D:\\study0_reorganized
data_SL\\subject*_TD_sharing_touch.csv"))))

#remove NaNs (not sure why they're read in)
td = td.dropna(how="all")

#label data by diagnosis
label = []
for x in asd['x']:
    label.append(1)
for x in td['x']:
    label.append(0)
```

```
#concatenate two datasets together
touch = pd.concat([asd,td], ignore_index=True)

#insert label column into dataset
touch.insert(0,"Label",label)

#tidy up
del label, x, asd, td

#sort by time
touch = touch.sort_values(['sessionId','time'])

# Visualise Current Features #############################################
plt.style.use('seaborn-white')
sns.set_context("talk")

touch.touchPhase.value_counts()
plt.figure("TP", figsize=(10,3))
plt.title("Distribution of Touch Phase Values")
plt.ylabel("Touch Phase Value")
plt.xlabel("Number of Occurences")
plt.hist(touch.touchPhase, color="gold", edgecolor="black", linewidth="2.5",
        orientation="horizontal")
filename = ("#TouchPhase.png")
plt.savefig(filename,format="png",bbox_inches="tight")

plt.figure("TC", figsize=(10,3))
plt.title("Distribution of Tap Count Values")
plt.ylabel("Tap Count Value")
plt.xlabel("Number of Occurences")
plt.hist(touch.tapCount, color="gold", edgecolor="black", linewidth="2",
        orientation="horizontal")
filename = ("#TapCount.png")
plt.savefig(filename,format="png",bbox_inches="tight")

plt.figure("TN", figsize=(10,3))
plt.title("Distribution of Touch Number Values")
plt.ylabel("Touch Number Value")
plt.xlabel("Number of Occurences")
plt.hist(touch.touchNumber, color="gold", edgecolor="black", linewidth="2",
         range=(0,5), orientation="horizontal")
filename = ("#TouchNumber.png")
plt.savefig(filename,format="png",bbox_inches="tight")

#plot heatmaps of touch ##################################################
touchtd=touch[touch.Label==0]
touchasd=touch[touch.Label==1]

plt.figure("TD", figsize=(10,6))
plt.title("Where TD Subjects Touch the iPad")
plt.ylim(0,800)
plt.hist2d(touchtd.x,touchtd.y,bins=[256,192], cmap="gnuplot")
plt.colorbar()
filename = ("#TD_Heatmap.png")
plt.savefig(filename,format="png",bbox_inches="tight")

plt.figure("ASD", figsize=(10,6))
```

```
plt.title("Where ASD Subjects Touch the iPad")
plt.ylim(0,800)
plt.hist2d(touchasd.x,touchasd.y,bins=[256,192], cmap="gnuplot")
plt.colorbar()
filename = ("#ASD_Heatmap.png")
plt.savefig(filename,format="png",bbox_inches="tight")

#tidy up
del filename, touchasd, touchtd
##########################################################################

#touchNumber variable is only ever 0, remove it and tapCount
touch = touch.drop(['touchNumber'], axis=1)
touch = touch.drop(['tapCount'], axis=1)
touch = touch.reset_index(drop=True)

##########################################################################
#Find start and end of swipes ############################################

#create new df without touchPhase = 1 data
touchredux = touch[touch.touchPhase != 1]

#initialise variables
x1,x2,y1,y2,t1,t2,id1,id2,ix1,ix2,lab = ([] for i in range(11))

#get list of all session IDs
IDs = sorted(touch.sessionId.unique())

#populate new variables with swipe data
#touchPhase == 0 means swipe started
#touchPhase == 3 or 4 means swipe ended
#record starting and ending positions and times

s=0
i=0
while s < len(IDs):
    trsub = touchredux.loc[touchredux.sessionId == IDs[s]]
    num = range(0,len(trsub))
    trsub.insert(6,"Num",num)

    while i < (len(trsub)-2):
        val = trsub.iat[i,4]
        nexval = trsub.iat[i+1,4]

        if (val == 0) and (nexval == 3 or nexval == 4):
            x1.append(trsub.iat[i,2])
            y1.append(trsub.iat[i,3])
            t1.append(trsub.iat[i,5])
            id1.append(trsub.iat[i,1])
            ix1.append(trsub.index[i])
            lab.append(trsub.iat[i,0])
            x2.append(trsub.iat[i+1,2])
            y2.append(trsub.iat[i+1,3])
            t2.append(trsub.iat[i+1,5])
            id2.append(trsub.iat[i+1,1])
            ix2.append(trsub.index[i+1])
            i+=2
```

```
        else:
            i+=1
    i=0
    s+=1

#create dataset for swipes
swipes = pd.DataFrame()

#insert label column into dataset
swipes.insert(0,"T1",t1)
swipes.insert(1,"T2",t2)
swipes.insert(2,"ID",id1)
swipes.insert(3,"IX1",ix1)
swipes.insert(4,"IX2",ix2)
swipes.insert(5,"X1",x1)
swipes.insert(6,"Y1",y1)
swipes.insert(7,"X2",x2)
swipes.insert(8,"Y2",y2)
swipes.insert(9,"Label",lab)

#tidy up
del t1,t2,id1,id2,i,nexval,val,ix1,ix2,x1,x2,y1,y2,lab,touchredux,s,trsub

#TouchPhase length
length=[]
for x in swipes.index:
    length.append((swipes.iat[x,4] - swipes.iat[x,3])+1)
swipes.insert(10,"Length",length)

#Plot Length Data
plt.figure("Swipe Length", figsize=(10,4))
plt.title("Distribution of Swipe Gesture Lengths")
plt.ylabel("Number of Occurences")
plt.xlabel("Length (TouchPhases)")
plt.hist(length, bins=100,color="gold",
         edgecolor="black", linewidth="2", range=(0,450))
filename = ("#Length.png")
plt.savefig(filename,format="png",bbox_inches="tight")
#tidy up
del filename
del x, length

#Count completed gestures for each cohort
ASDcount = []
TDcount = []
ASDcount.append(swipes.Label.value_counts()[1])
TDcount.append(swipes.Label.value_counts()[0])

#Remove any swipe lasting less than 10 touch phases
swipes = swipes[swipes.IX2 - swipes.IX1 > 10]
swipes = swipes.reset_index(drop=True)
swipes.Label.value_counts()
ASDcount.append(swipes.Label.value_counts()[1])
TDcount.append(swipes.Label.value_counts()[0])

##########################################################################
#Find touchphase=1 data for the defined swipes ###########################
```

```
#Create new column
Keep = []
for x in touch.index:
    Keep.append(0)
touch.insert(0,"Keep",Keep)
del Keep,x
#touch = touch.drop(['Keep'], axis=1)

#This takes a really long time to run, around 20 minutes
presub = 0
i=0
x=0
while i < len(IDs):
    tsub = touch.loc[touch.sessionId == IDs[i]]
    swsub = swipes.loc[swipes.ID == IDs[i]]
    p = tsub.index
    tsub = tsub.reset_index(drop=True)
    swsub = swsub.reset_index(drop=True)
    for j in tsub.index:
        for k in swsub.index:
            if tsub.iat[i,0] == 0:
                if (swsub.iat[k,3] == p[j]):
                    l = j + presub
                    touch.iat[l,0] = 1
                elif (swsub.iat[k,4] == p[j]):
                    l = j + presub
                    touch.iat[l,0] = 1
                if (swsub.iat[k,3] <= p[j] <= swsub.iat[k,4]) and (tsub.iat[j,5]
== 1):
                    l = j + presub
                    touch.iat[l,0] = 1
    presub = presub + len(tsub)
    print(i) #see how far along it is
    i+=1
#tidy up
del x,i,j,k,l,presub,swsub,tsub

#check
touch.groupby(["touchPhase", "Keep"]).size()

#delete "bad" readings and tidy up
touch2 = touch[touch.Keep == 1]
touch2 = touch2.drop(['Keep'], axis=1)
touch2 = touch2.reset_index(drop=True)

#save to pickle
touch2.to_pickle("touch2.pkl")

#plot swipes and save images ##############################################
i = 0
plt.style.use('seaborn-white')
while i < len(IDs):
    tsub = touch.loc[touch.sessionId == IDs[i]]
    t2sub = touch2.loc[touch2.sessionId == IDs[i]]
    plt.figure(i)
    plt.scatter(tsub.x,tsub.y,color="slategrey", s=20)
    plt.scatter(t2sub.x,t2sub.y,color="gold", s=30, alpha=0.5)
    if tsub.iat[1,1] == 0:
```

```
        label = "TD"
    else:
        label = "ASD"
    plt.title("Subject " + str(i) + " " + str(label))
    filename = ("Subject " + str(i) + " " + str(label) + ".png")
    plt.savefig(filename,format="png",bbox_inches="tight")
    plt.close()
    i+=1
del tsub,t2sub,i,filename,label

###########################################################################
#Identify which touch values belong to which swipes######################
swipenumber = []
num = 0
for x in touch2.index:
    if touch2.iat[x,4] != 3 and touch2.iat[x,4] != 4:
        swipenumber.append(num)
    elif touch2.iat[x,4] == 3 or touch2.iat[x,4] == 4:
        swipenumber.append(num)
        num+=1

touch2.insert(6,"SwipeID",swipenumber)
del swipenumber,num

ID = []
for x in swipes.index:
    ID.append(swipes.index[x])
swipes.insert(0,"SwipeID",ID)
del ID,x
###########################################################################
#Look for swipes containing multitouch data and discard them #############
i=0
mt=[]

#identify
while i <= max(touch2.SwipeID):
    swsub = touch2.loc[touch2.SwipeID == i]
    if any(swsub['time'].duplicated()):
        mt.append(i)
    i+=1

#discard
swipes=swipes.drop(swipes.index[mt])
swipes=swipes.reset_index(drop=True)
touch2=touch2[~touch2.SwipeID.isin(mt)]
touch2=touch2.reset_index(drop=True)
del i,swsub,mt

ASDcount.append(swipes.Label.value_counts()[1])
TDcount.append(swipes.Label.value_counts()[0])
###########################################################################
#Kinematic Calculations ##################################################

MinSpeed = []
Speed = []
MaxSpeed = []

MinAccel = []
```

```
Accel = []
MaxAccel = []

MinJerk = []
Jerk = []
MaxJerk = []

Distance = []

for i in swipes.SwipeID:
    tsub = touch2.loc[touch2.SwipeID == i]
    tsub = tsub.reset_index(drop=True)
    dist,vel,acc,jrk,tv,ta,tj,dur,abacc = ([] for r in range(9))
    j=1
    while j < len(tsub):
        x = (tsub.iat[j,2] - tsub.iat[j-1,2])
        y = (tsub.iat[j,3] - tsub.iat[j-1,3])
        dist.append(math.hypot(x, y))
        duration = tsub.iat[j,5]-tsub.iat[j-1,5]
        dur.append(duration)
        vel.append(math.hypot(x, y)/(duration))
        tvel = (tsub.iat[j,5]+tsub.iat[j-1,5])/2
        tv.append(tvel)
        j+=1
    k = 1
    while k < len(vel):
        duration = tv[k]-tv[k-1]
        abacc.append(abs((vel[k]-vel[k-1])/duration))
        acc.append((vel[k]-vel[k-1])/duration)
        tacc = (tv[k]+tv[k-1])/2
        ta.append(tacc)
        k+=1
    m = 1
    while m < len(acc):
        duration = ta[m]-ta[m-1]
        jrk.append(abs((acc[m]-acc[m-1])/duration))
        tjrk = ((ta[m]+ta[m-1])/2)
        tj.append(tjrk)
        m+=1

    MinSpeed.append(min(vel))
    Speed.append(sum(dist)/sum(dur))
    MaxSpeed.append(max(vel))

    MinAccel.append(min(abacc))
    Accel.append(np.mean(abacc))
    MaxAccel.append(max(abacc))

    MinJerk.append(min(jrk))
    Jerk.append(np.mean(jrk))
    MaxJerk.append(max(jrk))

    Distance.append(sum(dist))

#tidy up
del i,j,k,m,dist,vel,acc,jrk,tv,ta,tj,dur,duration,tvel,tacc,tjrk,tsub,x,y,abacc
```

```
#Calculate Duration
Duration = []
for i in swipes.index:
    Duration.append(swipes.iat[i,2] - swipes.iat[i,1])
del i

#add calculated values to Swipes Dataframe

swipes.insert(12,"MinSpeed",MinSpeed)
swipes.insert(13,"Speed",Speed)
swipes.insert(14,"MaxSpeed",MaxSpeed)

swipes.insert(15,"MinAccel",MinAccel)
swipes.insert(16,"Acceleration",Accel)
swipes.insert(17,"MaxAccel",MaxAccel)

swipes.insert(18,"MinJerk",MinJerk)
swipes.insert(19,"Jerk",Jerk)
swipes.insert(20,"MaxJerk",MaxJerk)

swipes.insert(21,"Distance",Distance)
swipes.insert(22,"Duration",Duration)

#tidy up
del
Accel,Jerk,Duration,Distance,Speed,MinSpeed,MaxSpeed,MinJerk,MaxJerk,MinAccel,MaxA
ccel

##########################################################################
#Define food to plate#####################################################

Food = []
Plate = []
Face = []

for i in swipes.index:
    if (233.5 <= swipes.iat[i,6] <= 772) and (121 <= swipes.iat[i,7] <= 265.5):
        Food.append(1)
    else:
        Food.append(0)

for i in swipes.index:
    if (94.5 <= swipes.iat[i,8] <= 919.5) and (265.5 <= swipes.iat[i,9] <= 332.5):
        Plate.append(1)
    else:
        Plate.append(0)

for i in swipes.index:
    if (94.5 <= swipes.iat[i,8] <= 919.5) and (332.5 <= swipes.iat[i,9] <= 520):
        Face.append(1)
    else:
        Face.append(0)

swipes.insert(23,"Food",Food)
swipes.insert(24,"Plate",Plate)
swipes.insert(25,"Face",Face)

del Food,Plate,Face
```

```
FtP=[] #Food to Plate
FtF=[] #Food to Face

for i in swipes.index:
    if (swipes.iat[i,23] == 1) and (swipes.iat[i,24] == 1):
        FtP.append(1)
    else:
        FtP.append(0)


for i in swipes.index:
    if (swipes.iat[i,23] == 1) and (swipes.iat[i,25] == 1):
        FtF.append(1)
    else:
        FtF.append(0)

swipes.insert(26,"FtP",FtP)
swipes.insert(27,"FtF",FtF)

del FtP,FtF,i

##########################################################################
# Find goal-oriented gestures ############################################
GoalOG = []
for i in swipes.index:
    if (swipes.iat[i,26] == 1) or (swipes.iat[i,27] == 1):
        GoalOG.append(1)
    else:
        GoalOG.append(0)

swipes.insert(28,"GoalOG",GoalOG)
GoalOG = swipes[swipes.GoalOG == 1]
GoalOG=GoalOG.drop(GoalOG.index[[2279]])
GoalOG = GoalOG.reset_index(drop=True)
del i

#Count Goal-Oriented Gestures
ASDcount.append(GoalOG.Label.value_counts()[1])
TDcount.append(GoalOG.Label.value_counts()[0])
#Count Food-to-Plate Gestures
ASDcount.append(GoalOG[GoalOG.FtP==1].Label.value_counts()[1])
TDcount.append(GoalOG[GoalOG.FtP==1].Label.value_counts()[0])
#Count Food-to-Face Gestures
ASDcount.append(GoalOG[GoalOG.FtF==1].Label.value_counts()[1])
TDcount.append(GoalOG[GoalOG.FtF==1].Label.value_counts()[0])

##########################################################################
## Plot Counts of Gesture Types ##########################################

#Calculte percentage of gestures from ASD group
percs = []
i=0
while i < len(ASDcount):
    percs.append(100*(ASDcount[i]/(ASDcount[i]+TDcount[i])))
    i+=1

#Plot
```

```
fig, ax = plt.subplots(figsize=(10, 5))
x = np.arange(6)    # the x locations for the groups
width = 0.2         # the width of the bars
TD = ax.bar(x, TDcount, width, color='gold',edgecolor="black")
ASD = ax.bar(x + width, ASDcount, width, color='tomato',edgecolor="black")
ax.set_title('Gesture Type Count by Diagnosis')
ax.set_xticks(x+0.1)
ax.set_xticklabels(('All Gestures', 'Taps Removed',
                    'Multi-Touch\nRemoved',"Goal-Oriented",
                    "Food-to-Plate","Food-to-Face"), rotation=45)
ax.legend((TD[0], ASD[0]), ('TD', 'ASD'))
plt.ylim(0,7000)
i=0
while i < len(ASDcount):
    plt.text(x[i]+0.15, ASDcount[i]+200, "%0.01f%%" %percs[i])
    i+=1
filename = ("#gesturecount.png")
plt.savefig(filename,format="png",bbox_inches="tight")
plt.close()
del ASD,TD,x,width,ASDcount,TDcount,i,percs, filename

##########################################################################
# Plot Gesture Distribution ###############################################

props = GoalOG.groupby(["Label", "FtP"]).size()

fig, ax = plt.subplots(figsize=(10, 5), subplot_kw=dict(aspect="equal"))
colors = ['#fff099', 'gold', "#ff7d66", '#ff5233']
explode = (0.03, 0.03, 0.03, 0.03)
wedgeprops = {'linewidth': 2, 'edgecolor' : "black"}
ax.pie(props, colors=colors, explode=explode,
       wedgeprops=wedgeprops,autopct='%1.1f%%')
plt.title('Distribution of Gestures by Type and Diagnosis', y=1.02, fontsize=25)
plt.axis('equal')
ax.legend(('TD Food-to-Face', 'TD Food-to-Plate',
          'ASD Food-to-Face', 'ASD Food-to-Plate'),
    loc="center left", bbox_to_anchor=(0.75, 0, 0.5, 1))
filename = ("#swipepie.png")
plt.savefig(filename,format="png",bbox_inches="tight")
plt.close()
del colors, explode, wedgeprops,filename,props

##########################################################################
#histograms

plt.hist(GoalOG.Speed,bins=100)
plt.hist(GoalOG.MaxSpeed,bins=100)
plt.hist(GoalOG.MaxAccel,bins=100)
plt.hist(GoalOG.Jerk,bins=100)
plt.hist(GoalOG.Distance,bins=100)

##########################################################################
#Box-Plots ###############################################################

variables = list(GoalOG)[11:23]

ylab = ["Length (Touch Phases)","Speed (Pixels/Second)",
        "Speed (Pixels/Second)","Speed (Pixels/Second)",
```

```
        "Acceleration (Pixels/Second$^2$)",
        "Acceleration (Pixels/Second$^2$)",
        "Acceleration (Pixels/Second$^2$)",
         "Jerk (Pixels/Second$^3$)","Jerk (Pixels/Second$^3$)",
         "Jerk (Pixels/Second$^3$)","Distance (Pixels)","Duration (Seconds)"]

i=0
while i < len(variables):
    plt.figure(figsize=(7,10))
    meanlineprops = dict(linestyle="--", color='tomato', linewidth=2.5)
    box = sns.boxplot(x="Label", y=variables[i], data=GoalOG,
                      color="gold",linewidth=2.5, width=.5, showmeans=True,
                      meanprops=meanlineprops, meanline=True,
                      showfliers=False)
    plt.title("Comparison of " + variables[i] + " Across Cohorts")
    plt.xlabel('Cohort')
    plt.ylabel(ylab[i])
    x = np.array([0,1])
    my_xticks = ['TD','ASD']
    plt.xticks(x, my_xticks)
    filename = ("##" + variables[i] + ".png")
    plt.savefig(filename,format="png",bbox_inches="tight")
    plt.close()
    i+=1
del filename, meanlineprops,my_xticks,ylab,i,x

np.median(GoalOG.MinSpeed[GoalOG.Label==0])
np.median(GoalOG.MinSpeed[GoalOG.Label==1])
##########################################################################
# MannWhitney U Test #####################################################

TD = GoalOG[GoalOG.Label == 0]
ASD = GoalOG[GoalOG.Label == 1]
i="X2"
mw = pd.DataFrame()
Pvals=[]
for i in variables:
    stat,p = scipy.stats.mannwhitneyu(TD[i],ASD[i], alternative="two-sided")
    Pvals.append(p)
mw["Variable"]=variables
mw["p"]=Pvals
del stat,p,i,variables,Pvals,TD,ASD

##########################################################################
#Plot Goal Oreinted Swipes ###############################################

touch3=touch2[touch2.SwipeID.isin(GoalOG.SwipeID)]

i = 0
plt.style.use('seaborn-white')
while i < len(IDs):
    tsub = touch.loc[touch.sessionId == IDs[i]]
    t3sub = touch3.loc[touch3.sessionId == IDs[i]]
    plt.figure(i)
    plt.xlim(0.5,1024.5)
    plt.ylim(0.5,768.5)
    plt.scatter(tsub.x,tsub.y,color="slategrey", s=20)
    plt.scatter(t3sub.x,t3sub.y,color="gold", s=30, alpha=0.5)
```

```
    if tsub.iat[1,1] == 0:
        label = "TD"
    else:
        label = "ASD"
    plt.title("Subject " + str(i) + " " + str(label))
    filename = ("G-O Subject " + str(i) + " " + str(label) + ".png")
    plt.savefig(filename,format="png",bbox_inches="tight")
    plt.close()
    i+=1
del tsub,t3sub,i,filename,label,IDs

##########################################################################
##########################################################################
#Modelling ###############################################################

#Create final dataframe to be fed to the model
#Remove irrelevent variables e.g. timestamps, sessionID and swipe number
SwipeData = GoalOG.drop(["SwipeID","T1","T2","ID","IX1","IX2","GoalOG",
                         "Food","Plate","Face","Distance","FtP","FtF"], axis=1)
SwipeData.to_pickle("SwipeData.pkl")

##########################################################################

#Create target variable and remove Label from dataframe
y=SwipeData.Label
SwipeData = SwipeData.drop(["Label"],axis=1)
X = SwipeData

#How is the data split between classes?
y.value_counts(True)

#Create training and test sets - 90/10 split
X_train,X_test,y_train,y_test = \
sklearn.model_selection.train_test_split(X, y, stratify=y,
                                        test_size=0.1, random_state=8)

#Unbalanced classes - Need to Oversample
ros = RandomOverSampler(random_state=8)
X_train_res, y_train_res = ros.fit_sample(X_train, y_train)

##########################################################################
# Decision Tree Classifier ###############################################

params = [{"max_depth":[5,10,15,20],"criterion":["gini","entropy"],
         "min_samples_split":[2,4]}]

DecTree = DecisionTreeClassifier()
DTgs = GridSearchCV(DecTree, params, cv=10, scoring="recall")
DTgs.fit(X_train_res, y_train_res)

print(DTgs.best_estimator_)
print(DTgs.best_params_)
print(DTgs.best_score_)

predicted = DTgs.best_estimator_.predict(X_test)
print(metrics.classification_report(y_test, predicted))
print(metrics.confusion_matrix(y_test, predicted))
print(metrics.roc_auc_score(y_test,predicted))
```

```
################################################################################
# Bagging ######################################################################

DTparams = DTgs.best_params_

Bparams = [{"n_estimators":[250,500,750,1000],"oob_score":[True],
            "base_estimator":[DecisionTreeClassifier(**DTparams)]}]

Bagging = BaggingClassifier()
Bgs = GridSearchCV(Bagging, Bparams, cv=3, scoring="recall")
Bgs.fit(X_train_res, y_train_res)

print(Bgs.best_estimator_)
print(Bgs.best_params_)
print(Bgs.best_score_)

Bpredicted = Bgs.best_estimator_.predict(X_test)
print(metrics.classification_report(y_test, Bpredicted))
print(metrics.confusion_matrix(y_test, Bpredicted))
print(metrics.roc_auc_score(y_test,Bpredicted))


################################################################################
# Random Forest ################################################################

RFparams = [{"n_estimators":[500],"oob_score":[False],
           "max_depth":[15],"criterion":["entropy"],
           "min_samples_split":[2]}]

Forest = RandomForestClassifier()
RFgs = GridSearchCV(Forest, RFparams, cv=5, scoring="recall")
RFgs.fit(X_train_res, y_train_res)

print(RFgs.best_estimator_)
print(RFgs.best_params_)
print(RFgs.best_score_)

RFpredicted = RFgs.best_estimator_.predict(X_test)
print(metrics.classification_report(y_test, RFpredicted))
print(metrics.confusion_matrix(y_test, RFpredicted))
print(RFgs.best_estimator_.feature_importances_)
print(metrics.roc_auc_score(y_test,RFpredicted))

#P-R curve
y_score = RFgs.predict_proba(X_test)[:,1]

precision, recall, _ = precision_recall_curve(y_test, y_score)

plt.figure(figsize=(7,7))
plt.step(recall, precision, color='black', where='post')
plt.fill_between(recall, precision, step='post', color='gold')

plt.xlabel('Recall')
plt.ylabel('Precision')
plt.ylim([0.0, 1.05])
plt.xlim([0.0, 1.0])
plt.title('Precision-Recall Curve for Random Forest Model')
filename="#PRcurve.png"
```

```
plt.savefig(filename,format="png",bbox_inches="tight")
plt.close()

##########################################################################
# The following code is adapted from
# https://towardsdatascience.com/fine-tuning-a-classifier-in-scikit-learn-
66e048c21e65
# and is not my work

from sklearn.metrics import roc_curve, precision_recall_curve, auc, make_scorer,
recall_score, accuracy_score, precision_score, confusion_matrix

def adjusted_classes(y_scores, t):
    """
    This function adjusts class predictions based on the prediction threshold (t).
    Will only work for binary classification problems.
    """
    return [1 if y >= t else 0 for y in y_scores]

def precision_recall_threshold(p, r, thresholds, t=0.5):

    # generate new class predictions based on the adjusted_classes
    # function above and view the resulting confusion matrix.
    y_pred_adj = adjusted_classes(y_scores, t)
    print(pd.DataFrame(confusion_matrix(y_test, y_pred_adj),
                       columns=['pred_neg', 'pred_pos'],
                       index=['neg', 'pos']))

y_scores = RFgs.predict_proba(X_test)[:,1]

p, r, thresholds = precision_recall_curve(y_test, y_scores)

#No false negatives
precision_recall_threshold(p, r, thresholds, 0.16)

#90% true positives
precision_recall_threshold(p, r, thresholds, 0.28)

##########################################################################
# Feature Importances ####################################################

for_importances = pd.DataFrame()
for_importances['Name'] = SwipeData.columns
for_importances['Value'] = RFgs.best_estimator_.feature_importances_
for_importances = for_importances.sort_values('Value',ascending=True)

x = np.arange(len(SwipeData.columns))
plt.barh(x,for_importances.Value, tick_label = for_importances.Name,
         align="center", color="gold",edgecolor="black")
plt.title('Feature Importances for Random Forest')
plt.xlabel('Importance')
plt.ylabel('Feature')
filename="#FeatureImportances.png"
plt.savefig(filename,format="png",bbox_inches="tight")
plt.close()

##########################################################################
# SVM ####################################################################
```

```
#normalise the data
scaler = StandardScaler()
X_std = scaler.fit_transform(X)

X_train,X_test,y_train,y_test = \
sklearn.model_selection.train_test_split(X_std, y, stratify=y,
                                        test_size=0.1, random_state=8)

#Unbalanced classes - Need to Oversample
ros = RandomOverSampler(random_state=8)
X_train_res, y_train_res = ros.fit_sample(X_train, y_train)


#RBF kernel
RBFparams = [{"C":[1,10,50,100],"class_weight":[{0:1,1:2}],
           "gamma":["auto",0.1,1,5],"kernel":["rbf"]}]

SVMmodRBF = sklearn.svm.SVC()
RBFgs = GridSearchCV(SVMmodRBF, RBFparams, cv=5, scoring="recall")
RBFgs.fit(X_train_res,y_train_res)

print(RBFgs.best_params_)
print(RBFgs.best_score_)

RBFpredicted = RBFgs.best_estimator_.predict(X_test)
print(metrics.classification_report(y_test, RBFpredicted))
print(metrics.confusion_matrix(y_test, RBFpredicted))
print(metrics.roc_auc_score(y_test,RBFpredicted))

#Linear
LINparams = [{"C":[1,10,50,100],"kernel":["linear"]}]

SVMmodLIN = sklearn.svm.SVC()
LINgs = GridSearchCV(SVMmodLIN, LINparams, cv=5, scoring="recall")
LINgs.fit(X_train_res,y_train_res)

print(LINgs.best_params_)
print(LINgs.best_score_)

LINpredicted = LINgs.best_estimator_.predict(X_test)
print(metrics.classification_report(y_test, LINpredicted))
print(metrics.confusion_matrix(y_test, LINpredicted))
print(metrics.roc_auc_score(y_test,LINpredicted))

#Polynomial
POLparams = [{"C":[1,10,50,100],"degree":[3,5,10],"kernel":["poly"]}]

SVMmodPOL = sklearn.svm.SVC()
POLgs = GridSearchCV(SVMmodPOL, POLparams, cv=5, scoring="recall")
POLgs.fit(X_train_res,y_train_res)

print(POLgs.best_params_)
print(POLgs.best_score_)

POLpredicted = POLgs.best_estimator_.predict(X_test)
print(metrics.classification_report(y_test, POLpredicted))
print(metrics.confusion_matrix(y_test, POLpredicted))
print(metrics.roc_auc_score(y_test,POLpredicted))
```