5-2021

Using Deep Learning for Children Brain Image Analysis

Rafael Toche Pizano

USING DEEP LEARNING FOR CHILDREN BRAIN IMAGE ANALYSIS

An Undergradaute Honors College Thesis

in the

Department of Computer Science and Computer Engineering
College of Engineering
University of Arkansas
Fayetteville, AR
April, 2021

by

Rafael Toche Pizano
Bachelor of Science in Computer Science 2021

This thesis is approved by:

Justin Zhan, Ph.D.
Thesis Advisor:

John M. Gauch, Ph.D.
Committee member

Brajendra Panda, Ph.D.
Committee member

Abstract

Analyzing the correlation between brain volumetric/morphometry features and cognition/behavior in children is important in the field of pediatrics as identifying such relationships can help identify children who may be at risk for illnesses. Understanding these relationships can not only help identify children who may be at risk of illnesses, but it can also help evaluate strategies that promote brain development in children. Currently, one way to do this is to use traditional statistical methods such as a correlation analysis, but such an approach does not make is easy to generalize and predict these how brain volumetric/morphometry will impact cognition/behavior. One of the cognition behaviors that can be predicted is the IQ score. In the age of artificial intelligence and machine learning, it has become fundamental to be able to exploit techniques such as deep learning to automate and improve tasks. One of the types of data that is used to make such assessments is mean diffusivity data (MD). In this paper I propose using a machine learning approach and use the MD data to predict the IQ score of healthy 8-year old children. These predictions will provide insight into how well MD represents the IQ score of healthy 8-year old children and they will allow experts better understand how a child's neuropsychological score is affected by volumetric/morphometry data. In this paper I examine five different neural network models for predicting the IQ score of healthy 8-year old children. Each model is different in either the architecture and how the data is processed before it is fed to into the neural network. After analyzing these models, I found that the best performing neural network for the data set we are working with consists of using the Principal Component Analysis (PCA) for feature reduction and standardization of the data. On average, this model's IQ score predictions deviate from the true IQ score by 8.09%. Given the small data set and the high dimensionality of the data, it is concluded that this model's IQ score predictions are reasonable and well modeled IQ scores for healthy 8-year old children.

TABLE OF CONTENTS

1 Introduction

This paper focuses on a study [1] that assesses how volumetrics and morphometry data correlates with the neurophsychological scores in healthy 8-year old children. The data acquisition process is descried in the paper [1] and this research project is a joint collaboration with UAMS and Dr. Xiawei Ou, Rajikha Raja, and Xiaoxu Na. This research project emphasizes using mean diffusivity (MD) data to predict the IQ scores of healthy 8-year old children. For this project, we will also assume the data collected was appropriately assessed.

Machine learning techniques, such as deep learning, have surged past academic settings and are being put to use in real world settings to solve real world problems. For example, Netflix uses recommendation systems to suggest their user base new shows or movies that they might enjoy based on their viewership-history [2]. Machine learning is also used to optimize learning and teaching experiences. Language learning is one prime example of this. Duolingo, a language learning application, has access to billions of statistics regarding language learning through their user base. This data has allowed Duolingo to create a new statistical model called *half-life regression (HLR)* to estimate the half-life of a word in a user's memory by analyzing their users' error patterns [3]. Another practical and more impactful application is Google's recent deep learning model that assists pathologists in detecting cancer in patients [4].

It is clear that machine learning techniques are practical for statistical analysis in different fields. However, building machine learning models can be a convoluted process as the correct statistical model, and hyperparameters must be meticulously selected to create an efficient and reliable model. A machine learning model for learning, such as the Duolingo model mentioned above, is usually not applicable outside its specified domain, therefore, each problem that is to be tackled using a machine learning approach must be researched and properly understood to create the most efficient and reliable model possible.

In this thesis I will analyze different machine learning models and techniques to predict the neuropsychological scores of healthy 8-year old children using volumetrics and morphometry data. We will be using data provided by Dr. Xiawei Ou at the University of Arkansas Medical School (UAMS).

2 Related Works

2.1 Working with Small Medical Data Sets

Machine learning has found its way into the medical field due to its ability to find numerical correlations between features faster than a human as it aids physicians when making diagnoses. My research focuses on brain volumetrics and morphometry data on healthy 8-year old children and although there exists no prior work that involves volumetrics and morphometry data for healthy 8-year old children (excluding the paper this research project is based on [1]) we can use similar approaches to try and tackle our problem at hand. One of the first and most challenging problems about working with medical data sets is that they come in relatively small sizes. For example, the data set we have for this particular project contains volumetrics and morphometry data for only 72 patients. This can be a problem as machine learning models are usually used in contexts where larger data sets are available. It has been shown that small or insufficient training data can negatively impact the performance of a machine learning algorithm [5].

Thankfully, there have been a number of research projects that have aimed at tackling this problem. It has been shown that regression models such as neural networks can have a high accuracy for for data sets as small as 35 samples. In [6], the authors used a neural network with a data set of physico-mechanical properties of trabecular bone of only 35 samples and they were able to reach a regression factor of 0.96. This allowed the researches to asses that age is a factor in how the physical properties of the bone is affected by severe osteoarthritis [6]. In another study [7], it is discussed that bio-engineering data sets are usually small in size. The authors of this research project were able to craft a neural network for osteoarthritic bone fracture risk stratification and reached an accuracy of high

98.3% [7]. This biomedical engineering research project used focused on leveraging multiple runs of different neural network configurations to determine the most optimal architecture for their goals [7]. This research project demonstrates that small data sets are not necessarily detrimental for a machine learning model. However, it is important to note that the data used in this study consisted of only 5 features per sample, thus the dimensionality of their data set was not a problem. Our data for volumetrics and morphometry alone contains 279 features per sample. The implication is as follows: as the dimensionality of a data set grows, the amount of data to create reliable statistical analyses will grow exponentially [8]. This is otherwise known as ”the curse of dimensionality”.

2.2 PCA for High Dimensional Data

One of the known methods to reduce the dimensionality of a data set is called the Principal Component Analysis (PCA). The PCA is a popular approach to dealing with high dimensional data as it is able to reduce the dimensionality of a data set while at the same time reducing information loss [9]. This allows for easier interpretability of the data and in many cases, visualization. The data is transformed into a feature space where a smaller feature space represents the original feature space through the creation of uncorrelated features whose variance has been maximized [9]. The PCA's uses and effectiveness has been used in different applications to reduce dimensionality and perform a more thorough analysis of the data at hand It has been used in medical image processing with for feature extraction [10], noise reduction in high spectral resolution atmospheric sounding data [11], and image classification [12]. Because the PCA has been applied in different applications with resilient results, we will use PCA as one of our methods in our attempt to build a machine learning model to predict the neuropsychological scores of healthy 8-year old children.

2.3 Previous Analyses on Intelligence Using Brain Imaging Data

Analyzing brain image date to asses and predict intelligence features is not new. Multiple studies have attempted to use different types of brain imaging data and find a correlation between between them and intelligence scores such as the intelligent quotient (IQ).

One example of these studies is the study conducted by Liye Wang Et al. titled "MRI-Based Intelligence Quotient (IQ)Estimation with Sparse Learning" [13]. This study primarily focuses on a novel approach for estimating IQ scores based on Magnetic Resonance Imaging (MRI) data. Their implementation focuses on using a multi-kernel Support Vector Regression (SVR) and a single-kernel SVR in order to estimate the IQ scores. Their novel approach yields a root-mean square error of 8.695 [13].

The study mentioned above is not the only of its kind. Agoston Mihalik Et al. focused on applying a probabilistic segementation and kernel ridge regression to predict individual fluid intelligence based on T1-weighted MRI scans as apart of the ABCD Neurocognitive Prediction Challenge in 2019 [14]. Their appraoch of using a Kernel Ridge egression yielded a mean-squared error of 69.7204 [14] which placed the group in fifth place on the validation leaderboard and first place on the test leader board [14].

A study that relates more to our work was conducted by Kirsten Hilger Et al. where brain gray matter volume was used to predict IQ scores using machine learning models. The data used for this analysis was voxel-based morphometry [15]. Their goal was to determine whether whether region-specific anatomical differences were responsible for the overall intelligence or an individual, however, their models' mean absolute errors were significantly high, by more than ten IQ points [15]. This lead the team to conclude that general intelligence may not be attributed to specific regions in gray matter volume [15].

These studies show us that there is still much work left to be done when it comes to brain image data analysis for either the evaluation of intelligence or the prediction of intelligence factors such as IQ. Different methodologies and data can help us build better predictive models that allow us to better understand how intelligence factors relate to specific types of brain image data.

3 Preliminaries

Before we delve into the implementation of our machine learning model for predicting IQ scores, we must first discuss and clarify what the input and outputs of our model. Since the data shared with us by UAMS cannot be freely disclosed due to proprietary rights, we will simply give an overview of the type of data we will be working with and some basic general features about the data. It is also important to clearly state what our methodology will be and the types of approaches that will be taken to create the predictive models.

3.1 An Explanation of the Data and Its Uses

Although UAMS gave us five data sets to initially work with, our focus for now was on using one particular data set to predict IQ scores. We will be building our model based on information provided by the mean diffusivity (MD) data set. Mean diffusivity, or MD, characterizes the mean-squared displacement of molecules to diffusion [16]. The MD data set contains 66 samples and 68 features per sample, where each sample in the MD data set corresponds to a healthy 8-year old child. We will use a separate data set that contains the neuropsychological scores for all healthy 8-year children who were tested. However, it is important to note that the data containing the neuropsychological score contains data for 81 subjects whereas the MD data contains data set contains data for 66 subjects. This means that out of the 81 subjects for whom we have neuropsychological test scores, 15 of those subjects will not have MD data. This means that in total, we will have both MD and neuropsychological test scores data for only 66 subjects. The inputs of our predictive model will be the MD data and the outputs will be one particular feature of the neuropsychological data set, the IQ score.

This gap in the data sets is due to certain subjects not showing up for the later parts of the study conducted by X. Ou Et al. [1]. Decreasing the size of the data makes it more challenging to build a robust predictive model as fewer data means that the predictions will be less accurate and our model will be limited in terms of error minimization. However, since medical data tends to come in small sizes, our goal is to try and extract as much information as possible from the data available and create predictive models that can give us a good estimate of neuropsychological scores, in our case IQ, given the MD data.

3.2 A General Overview of the Models

For this research project, I will be using one general model, a neural network (NN). However, the architecture and hyperparameters of the model will differ from model to model in search of the best performing model. As a reminder, the data set being used for this project is small in sample size but high dimensional. Therefore, we are testing different models in an attempt to find the best performing model either by feature reduction or by hyperparameter modifications.

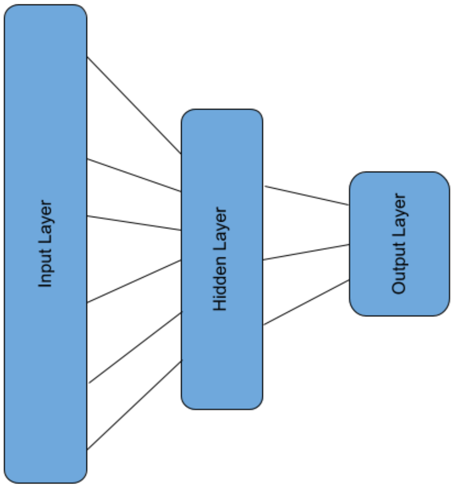


Figure 3.1: The general structure of our neural networks

Figure 3.1 above gives a general overview of what each of our models will look like. We will have an input layer, a hidden layer, and an output layer. As a reminder, in a neural network, the input layer will have n neurons. The number of neurons will match the dimensionality (number of features) of our data. The hidden layer will have have an 'arbitrary' number of neurons, however, this number will not be completely arbitrary. For some of our models we will use the proposed number of neurons by Jeff Heaton [17]. The number of neurons used and the method will be discussed in their respective sections. An introduction to the models that were implemented and a brief description of the has been included below.

> **Basic NN Architecture:** The first model that will be tested using the MD data provided by UAMS will be a simple neural network architecture. Our goal is to use this network and compare the other models to this one to show that the proposed modifications and data pre-processing yield a better result than a 'vanilla' architecture. No data pre-processing is applied here because the MD input data is between 0 and 1. Therefore, normalizing the data would not do much in this case. However, we will explore different data pre-processing techniques.
>
> **A NN and Breaking Data into Percentiles:** The second model will follow the architecture of the the first (a basic neural network), however, we will break up the data into 2 smaller data sets. Since we will be predicting the IQ score using the MD data, we will run a simple statistical analysis on the data and subset the data that contains our neuropsychological scores and group all healthy 8-year old children who had an IQ score lower than 50th percentile score, and those who had an IQ score greater than the 50th percentile score.
>
> **NN and Using A Correlation Matrix for Feature Reduction:** For this neural network, we will modify some of the hyperparameters and, in an attempt to increase the accuracy of the model, a correlation matrix will be used to identify those features that have high correlation and remove one of the features. The idea is that only those feature that represent most of the

data relevant information will be left and the model will yield better IQ score estimations.

NN, Correlation Matrix and Standardization: This approach is very similar to the one explained above, using a a correlation matrix for feature reduction. However, after applying the feature reduction technique, the data will be standardized before running it through the neural network. Ideally, this should yield better approximations for the IQ score.

NN, PCA, and Standardization: For this neural network, we will use the principal component analysis (PCA) for feature reduction and then the data will be standardized. The idea behind this approach is that the PCA is a better approach dimensionality reduction and this in turn should convert the features into components that preserve the most relevant information (variability).

4 Implementation of the Models

Now, we will take a closer look at our models and their implementations. To recap, our models will all be neural networks that will have some input layer, hidden layer, and output layer. Although our general definition of our neural network models in figure 3.1 contains only one hidden layer, the neural networks may have more than one hidden layer. As stated in the previous section, the input layer for the models will differ in the number of neurons, however, every neural network model will only have one neuron for the output layer. This is because we are always only predicting one value, the IQ score.

4.1 Basic NN Architecture

The first model that was created was a simple neural network architecture. This model consisted of four layers: one input layer, two hidden layers, and one output layer. The first layer consisted of 68 neurons for the 68 features of our data set. Meaning, for this model there was no feature reduction. The two hidden layers in our model have 46 neurons and 31 neurons, respectively. The output layer consists of only one output neuron.

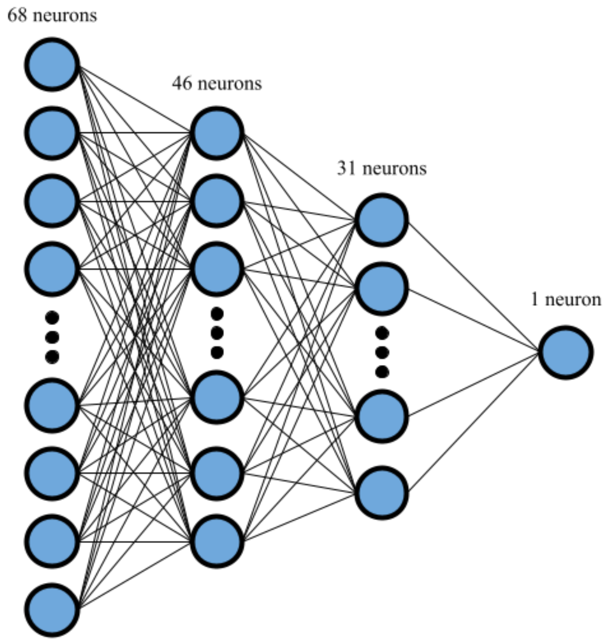


Figure 4.1: Basic NN Architecture

Although the number of neurons may seem arbitrary, I have followed the proposed number of neurons per hidden layer as outlined by Heaton [17], where he states that ”The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer”. Note that the model was fed all 68 features of the data. The MD and neuropsychological scores data were paired based on the patient IDs and 70% of the data was used as training data and the remaining was used as validation data. This model was trained over 1,000 epochs using the Adam optimizer, and the Mean Squared Error (MSE) as the criterion. The definition for MSE is defined as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

The advantage of using the MSE for this model was that highly inaccurate predictions will be highly penalized while ensuring that our predictions are also not outliers with too much error introduced into them.

The activation function used for the hidden layers was the Rectified Linear Units (ReLu). I decided to use the ReLu activation function for the models because of the results presented by Koutsoukas Et al. [18] where it was concluded that the ReLu activation function has an overall better performance than the Sigmoid or Tanh activation functions.

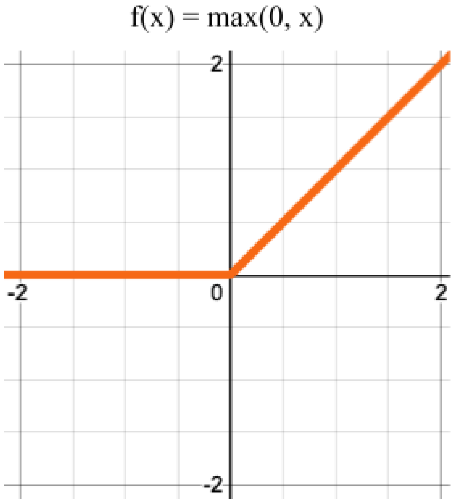
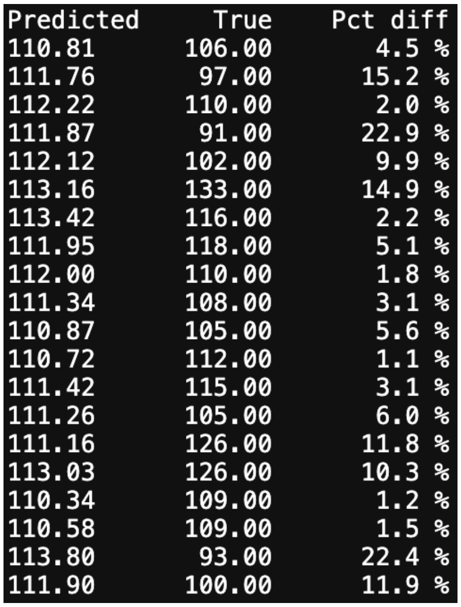


Figure 4.2: Representation of the ReLu activation function.

4.1.1 Evaluating the Results

Now that the architecture of the model and hyperparameters for the model have been stated, predictions of the model can be analyzed. After training the model for 1,000 epochs, the model yielded the following results. The *Predicted* column represents the predicted IQ scores by the neural network and the *True* column represents the true IQ score. The *Pct diff* column represents the percent difference between the predicted and true IQ scores.

The final training MSE loss for the model above was 187.16 and the final testing MSE loss was 111.22. The average absolute percent difference for the model was



```
Predicted      True    Pct diff
110.81       106.00       4.5 %
111.76        97.00      15.2 %
112.22       110.00       2.0 %
111.87        91.00      22.9 %
112.12       102.00       9.9 %
113.16       133.00      14.9 %
113.42       116.00       2.2 %
111.95       118.00       5.1 %
112.00       110.00       1.8 %
111.34       108.00       3.1 %
110.87       105.00       5.6 %
110.72       112.00       1.1 %
111.42       115.00       3.1 %
111.26       105.00       6.0 %
111.16       126.00      11.8 %
113.03       126.00      10.3 %
110.34       109.00       1.2 %
110.58       109.00       1.5 %
113.80        93.00      22.4 %
111.90       100.00      11.9 %
```

Figure 4.3: Results of the NN architecture

7.83%. At first glance, it seems like the model performed phenomenally, however, a closer look at the predicted values will explain why average absolute percent difference is so low. The average IQ score for the training data was 111.63 and most of the predicted IQ scores are very close to this value. This tells us that the model has a very low average percent difference because the model is over fitting. This is also explained by the nature of the size of the data set. There are not enough samples for the model to learn appropriately and the lack of outliers does not help on penalizing the model as it is learning. Therefore, as the model learns over 1,000 epochs, MSE loss function is converging the model to the average of the training set.

Figure 4.4 gives an overview of the distribution of the predicted IQ scores and the true IQ scores. The distribution of IQ scores helps visualize how the model was making very over fitted predictions. Overall, it is clear that the model makes unreliable predictions, even if they seem accurate. It is important to note although

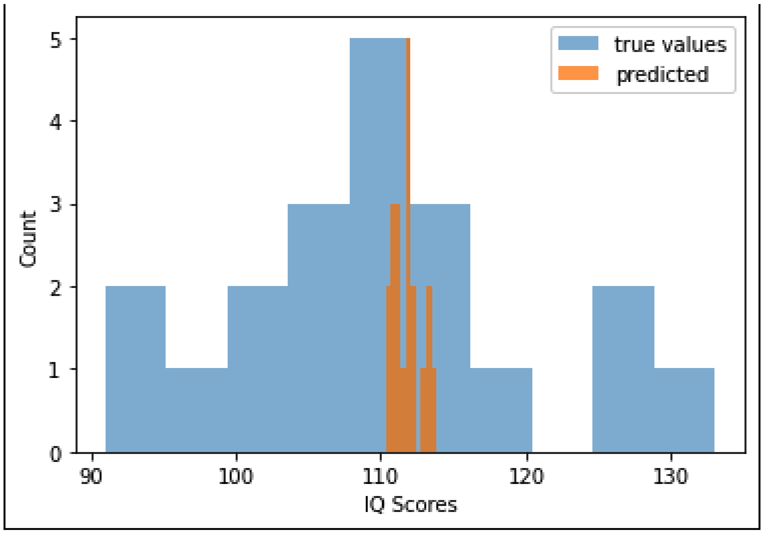


Figure 4.4: Distribution of predicted and true IQ scores.

this approach may work for data sets that have 10 times more samples than our data set, our model cannot learn what features influence the IQ score the most because of the high dimensional data and the small number of samples.

4.2 A NN and Breaking the Data into Percentiles

The second model that was created is very similar to the first model. However, there are some small differences that will be pointed out. In terms of the model itself, we use a neural network with the same amount of input features as the first, 68 features, but we will use only one hidden layer. The hidden layer will follow the number of neurons per hidden layer proposed by Heaton [17]. This means that with a 68 neuron input layer, the hidden layer will contain 46 neurons. As stated previously, the output layer will only contain one neuron for the prediction of the IQ score. The model will use the Adam optimizer, as well as the MSE as the criterion. The model will be trained over 500 epochs as this model will have to learn from an even smaller data set since we are breaking up our data set into two smaller sets, all IQ scores above the 50th percentile and all IQ scores below the 50th percentile.

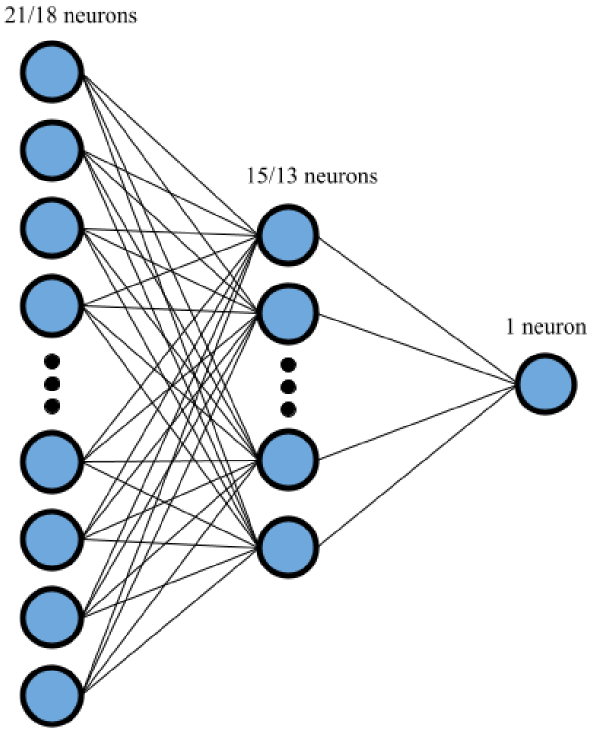


Figure 4.5: Model to train with data above/below 50th percentile

As stated previously, the data will be split up into two smaller sets in order that will represent all IQ scores above the 50th percentile and all IQ scores below the 50th percentile. However, before we can do this, we must first verify that the IQ scores are normally distributed. This approach only makes sense if we are working with data that follows a Gaussian distribution, therefore, we will use the empirical rule to verify that we are in fact dealing with normal IQ scores. The empirical has many uses, including calculating the probability of randomly observing a given variable, given that the variable comes from a normal distribution [19]. The process to use the empirical rule is as follows:

1. Convert the data to a standard score.

2. Verify that about 68% of the data is within 1 standard deviation from the mean.

3. Verify that about 95% of the data is within 2 standard deviations from the mean.

4. Verify that about 99.7% of the data is within 3 standard deviations from the mean.

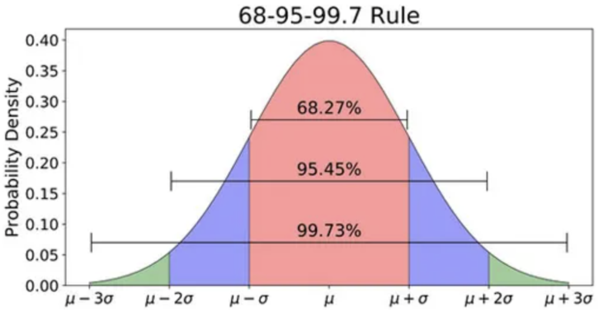


Figure 4.6: Visualizing the empirical rule. Image taken from Dr. McLeod [19]

Per Dr. Saul McLeod, the empirical rule tells one that data is normally distributed if it follows points 2-4 above. The empirical rule is also referred to as the three sigma rule because it is often defined as the 68-95-99.7 rule [19]. Figure 4.7 represents the distribution of the IQ scores, and the probability density function (PDF) of the IQ scores has been added for easier interpretability.

In order to compute the whether the IQ scores data follows a normal distribution, we will need to first understand the PDF for continuous random variables. The variables μ and σ to represent the sample mean and sample standard deviation, respectively. The PDF represents the distribution of probability of observing a value less than some random variable X [20]. The goal is to use the PDF to figure out whether the IQ scores follow a random distribution. In order to accomplish this, the upper and lower bound Z-scores are first computed and then they are used

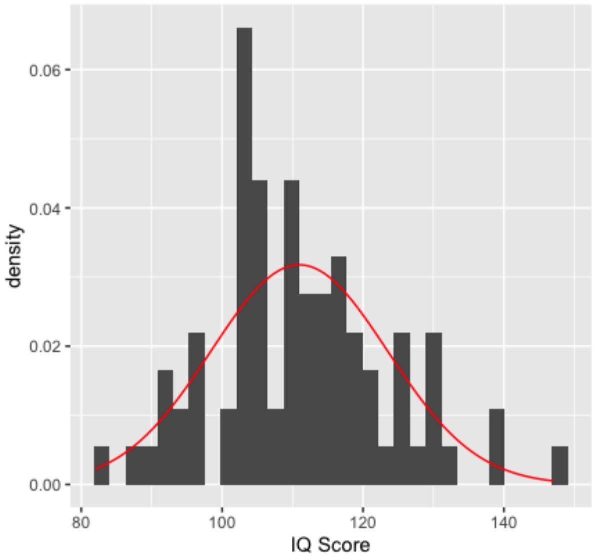


Figure 4.7: Distribution of IQ scores.

to find what percentage of the data lies between those two points. The formula for the PDF is as follows:

$$\int_{Z_{lower}}^{Z_{upper}} \frac{1}{\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}, -\infty < x < \infty$$

The IQ scores have a sample mean of $\mu = 110.95$ and a standard deviation of $\sigma = 12.56$. Using these two pieces of information, we can now use the R programming language to find if our data is normal based on the empirical rule.

Percent of data is <u>one</u> σ from μ? The lower and upper IQ scores one standard deviation away from the mean are 98.39 and 123.50, respectively. Calculating how much data is in between these two values yields a value of .679, which we can choose to round to .68. Thus, 68% of the data is within one standard deviation form the mean.

Percent of data is <u>two</u> σ from μ? The lower and upper IQ scores two standard deviations away from the mean are 85.83 and 136.02, respectively.

Calculating how much data is in between these two values yields a value of .9506, which we can choose to round to .95. Thus, 95% of the data is within two standard deviations form the mean.

Percent of data is three σ from μ? The lower and upper IQ scores three standard deviations away from the mean are 73.28 and 148.62, respectively. Calculating how much data is in between these two values yields a value of 1. In other words, 100% of the data is within three standard deviations form the mean. This is close enough to the the third rule of the empirical rule that states that 99.7% of the data is within 3 standard deviations from the mean.

Because the data is is normally distributed, we can go ahead and split the data into two subsets, all data above and below the 50th percentile. This results in two data sets:

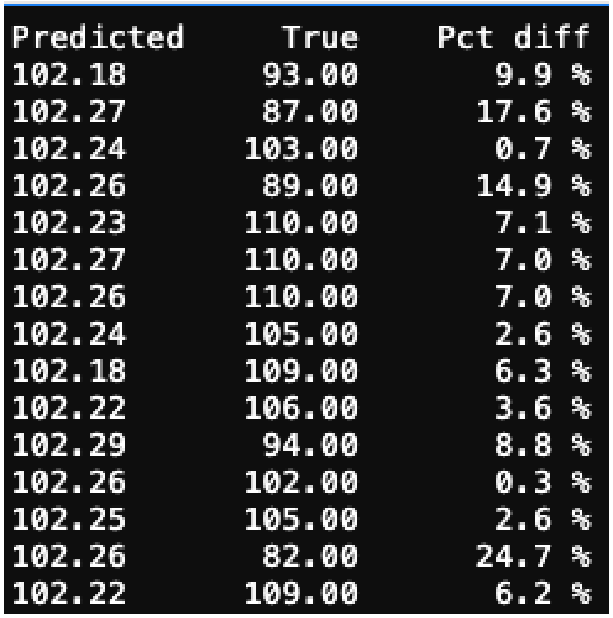
Data Below 50th Percentile: 36 samples where 60% is used as training data and the rest as testing data.

Data Above 50th percentile: 30 samples where 60% is used as training data and the rest as testing data.

The idea behind this approach is that since we are using a small data set, dividing the data into these two percentiles will allow the model to learn the most important features of each data set (below and above the 50th percentile). The predictions should be more representative of their respective percentiles, as opposed to the first model introduced that was over fitted to predict the average IQ score. Because we have a small data set and because we are breaking into two smaller data sets, it is expected that these models will be over fitted as well. However, evaluating these models is important as they will be compared to the future models.

4.2.1 Evaluating the Results for the Predictions Below the 50th Percentile

As stated previously, the model was trained over 500 epochs using the data that is below the 50th percentile.



```
Predicted        True     Pct diff
102.18          93.00        9.9 %
102.27          87.00       17.6 %
102.24         103.00        0.7 %
102.26          89.00       14.9 %
102.23         110.00        7.1 %
102.27         110.00        7.0 %
102.26         110.00        7.0 %
102.24         105.00        2.6 %
102.18         109.00        6.3 %
102.22         106.00        3.6 %
102.29          94.00        8.8 %
102.26         102.00        0.3 %
102.25         105.00        2.6 %
102.26          82.00       24.7 %
102.22         109.00        6.2 %
```

Figure 4.8: Predictions below the 50th percentile

Figure 4.8 shows how the model is predicting IQ scores for IQ scores below the 50th percentile. The average absolute error for the model is 7.95%. The final MSE loss for the validation samples was 85.04. Although the model has a very low absolute percent difference for the predictions, it is easy to see that it is over fitting. The predictions for the IQ score are all very close to 102, and although there is slight variation in the decimal part of the IQ score predictions, it is clear that the model is predicting the average IQ score below the 50th percentile. The slight variation in the decimal part of the predictions tells us that the model is learning from the data, however, the number of samples (36) that is being fed into the model is not enough.

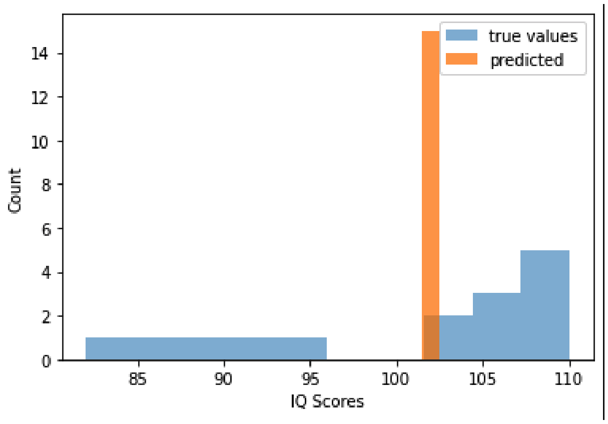


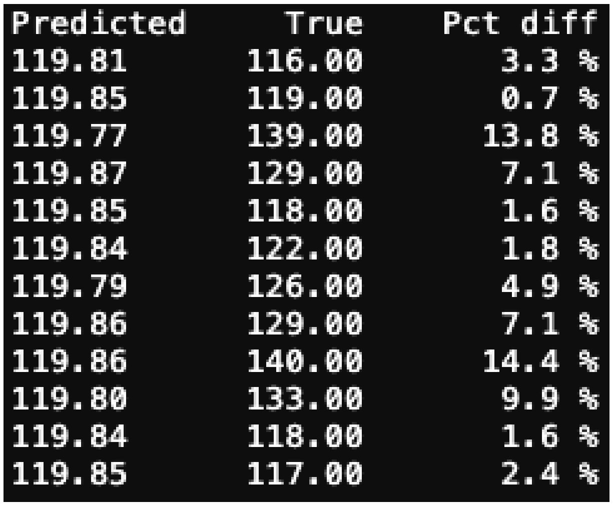
Figure 4.9: Distribution of predicted and true IQ scores.

Figure 4.9 helps visualize how the true and predicted distributions vary. As stated previously, the model has over fitted the data below the 50th percentile and is predicting an IQ score that is an average of the IQ scores below the 50th percentile. However, the IQ scores below the 50th percentile vary from about 85 to 110. Virtually, we can say that the IQ score predictions are exclusively 102.

4.2.2 Evaluating the Results for the Predictions Above the 50th Percentile.

The architecture that makes the predictions for the IQ scores above the 50th percentile is the same as the architecture that makes the predictions for the IQ scores below the 50th percentile. The only difference between the two models is the data that is being fed into them. That is, we are feeding into the model the input data that corresponds to the IQ scores above the 50th percentile.

Figure 4.10 shows the predicted IQ scores versus the true IQ scores and their respective percent differences. The average absolute percent difference is 5.71%. The validation MSE loss was 99.02. As in our result for the model that predicted IQ scores below the 50th percentile, it is clear that the model is over fitted to the data above the 50th percentile. Again, the model predictions are the average of the IQ



```
Predicted      True    Pct diff
119.81       116.00       3.3 %
119.85       119.00       0.7 %
119.77       139.00      13.8 %
119.87       129.00       7.1 %
119.85       118.00       1.6 %
119.84       122.00       1.8 %
119.79       126.00       4.9 %
119.86       129.00       7.1 %
119.86       140.00      14.4 %
119.80       133.00       9.9 %
119.84       118.00       1.6 %
119.85       117.00       2.4 %
```

Figure 4.10: Predictions for data above the 50th percentile.

scores above the 50th percentile. However, just as in the previous implementation, there is some variation in the decimal of the predictions. This indicates that the model has learned from the data it was fed. However, the amount of samples (30) that was fed to the model is not enough and more data would help the model make more reliable predictions.

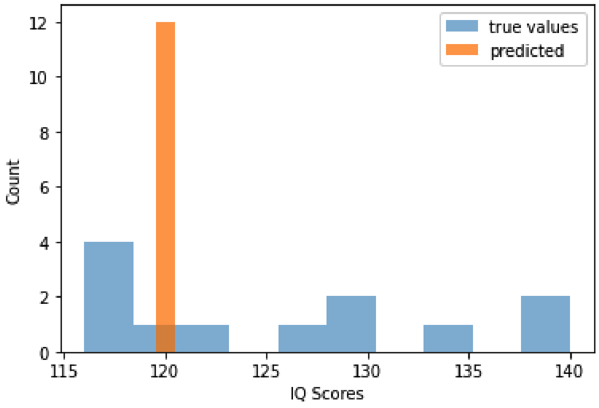


Figure 4.11: Distribution of predicted and true IQ scores.

Figure 4.11 gives an overview of the distribution of the predicted and true IQ scores. It is easy to see that virtually all IQ score predictions were 119, but the true IQ scores range from just above 115 to just below 140.

4.3 A NN and Using a Correlation Matrix for Feature Reduction:

The third model that was implemented during this research project follows a very similar architecture to that of the first two, however, the data pre-processing is different in that we attempt a feature reduction technique to improve the reliability and generalization of the model. The idea behind this approach is that reducing the number of features while keeping the most influential ones will result in a neural network that is able to learn better from the data than the first two, and make more generalized and reliable predictions instead of over fitting as the first two did. As mentioned earlier, the data set we are dealing with is small in sample size but large in dimensionality, therefore, reducing the number of features we feed into the neural network should allow the model to make more insightful predictions.

For this model, a correlation matrix will be used to find the most highly correlated features, and one of them will be dropped in an attempt to remove redundant information, thus reducing the amount of noise that the data contains. This approach was inspired through its uses in omics data as outlined by Yasset Perez-Riverol Et al. [21]. In this paper, it is discussed that a simple yet powerful approach is to remove feature redundancy is to use a correlation matrix filter [21]. For this approach, a correlation coefficient 0.85 or more was considered to be highly-correlated. This correlation coefficient was chosen in an attempt to get rid of highly correlated features and the logic as to why this exact cutoff (0.85) was chosen will be discussed later. As mentioned previously, the data we are working with consists of 68 features and the goal here is to reduce the number of features that will be fed into the model by dropping one of the features that belong to the set of features that have a correlation coefficient of 0.85 or more.

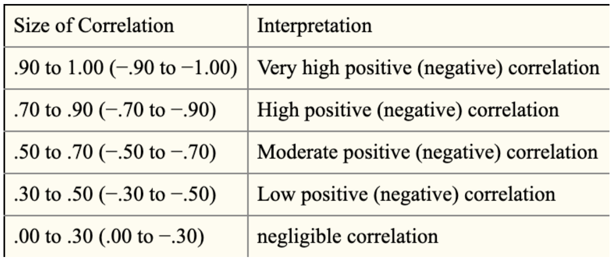
The method that we will use to find features that have a correlation of 0.85 or more is the Pearson Correlation Coefficient (PCC). The PCC is is a feature with no specific dimension that quantifies covariance, which ranges from +1 to -1, inclusive [22]. The values +1 and -1 represent the positive and negative correlation that exsits between two variables, respectively [23]. The PCC is defined as follows [24]:

r = correlation coefficient
x_i = values of xth term in variable X
$\bar{x}$ = mean of values in variable X
y_i = values of yth term in variable Y
$\bar{y}$ = mean of values in variable Y

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

It is important to note that the variable r used in the definition of PCC above is is often used to describe the mentioned relationship amongst two continuous random variables [25].

In order to eliminate those features which are considered to be redundant and, therefore, unnecessary, a PCC of 0.85 or more was chosen. This specific value may seem arbitrary but there is a reason as to why this value was chosen. It has often been discussed that in research that certain PCCs are inarguably 'strong' or 'negligible. For example, it has been discussed that PCCs of less than 0.1 can be considered to be negligible while PCCs or more than 0.90 are very strong [22]. Although there is no established method to concretely interpret these correlation coefficients, a rule of thumb can be used to interpret the correlation coefficients. Figure 4.12 shows the rule of thumb to be used when interpreting correlation coefficients, taken from [26].



Size of Correlation	Interpretation
.90 to 1.00 (−.90 to −1.00)	Very high positive (negative) correlation
.70 to .90 (−.70 to −.90)	High positive (negative) correlation
.50 to .70 (−.50 to −.70)	Moderate positive (negative) correlation
.30 to .50 (−.30 to −.50)	Low positive (negative) correlation
.00 to .30 (.00 to −.30)	negligible correlation

Figure 4.12: Rule of Thumb for the interpretation of correlation coefficients, taken from [26].

Using figure 4.12 as our basis for interpreting correlation coefficients and given that we want to get rid of as many redundant features as possible, the correlation coefficient of 0.85 was chosen as it is close to the 'very high positive correlation' and within the 'high positive correlation' rule.

With the logic behind this approach in place, the method that was used to reduce the number of features can be explained. The first step is to find the feature pairs that have PCC of 0.85 or greater. Once these feature pairs have been identified, one of them will be removed from the data set in, this is the feature reduction part of this approach.

Figure 4.13 gives an overview of the feature pairs in the MD data set that a PCC of 0.85 or greater. Although it is difficult to see how many feature pairs are at or above this PCC, it is easy to see that there in fact are feature pairs that have a PCC of 0.85 or more. To get a better understanding of the figure 4.13, an analysis was performed on the correlation matrix in figure 4.13 and a total of 78 feature pairs were found. Of these 78 feature pairs, it was found that 28 of them were unique and all other features correlated with these features. These 28 unique fea-

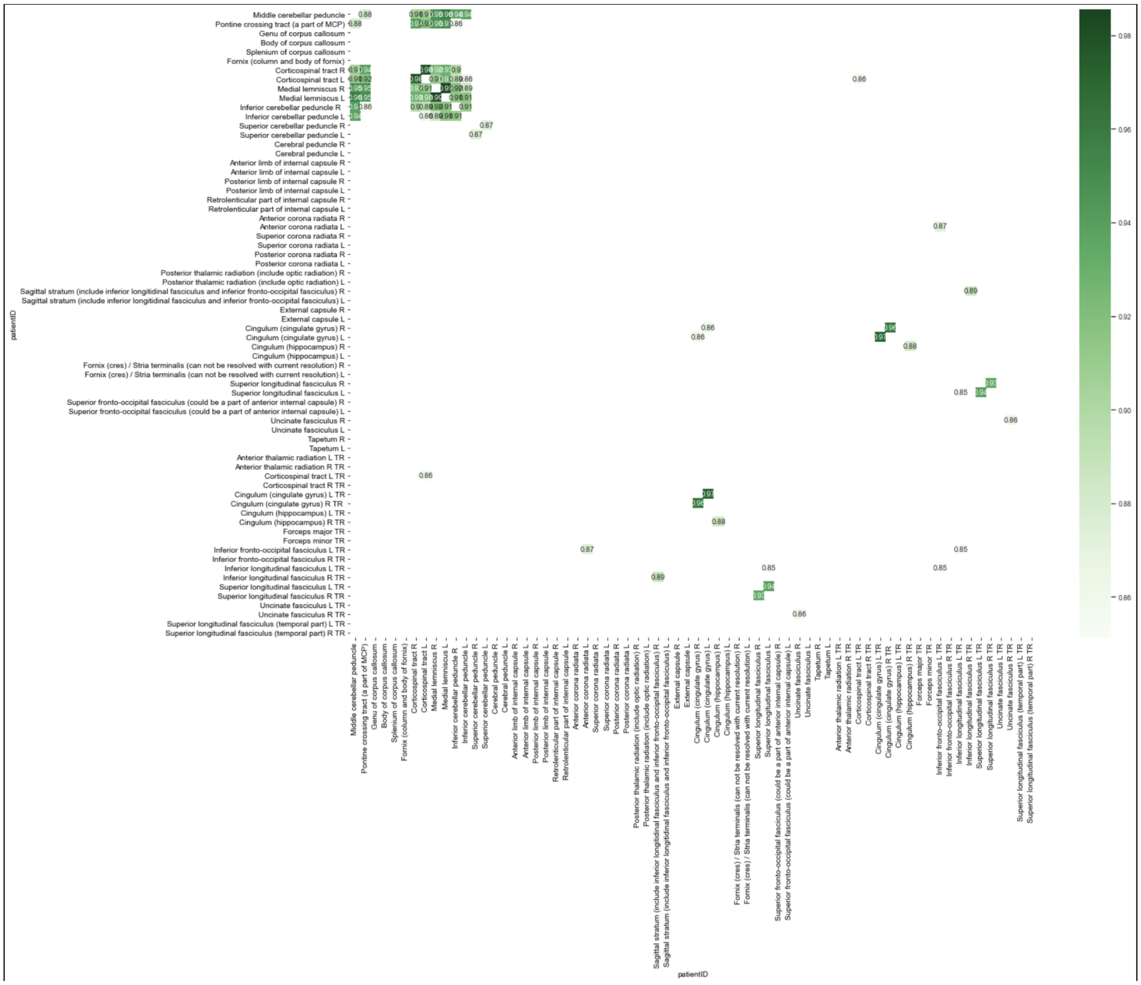


Figure 4.13: All features that have PCC of 0.85 or greater.

tures are the features that will be dropped from the data set.

Dropping these features creates a new data set with a smaller dimensionality than the original data set. The features in this new data set will have a correlation of less than 0.85. Figure 4.14 gives an overview of the correlations between the remaining set of features in the MD data set after dropping the 28 features mentioned above. Although it may be difficult to read due to the size of the correlation matrix, no feature pairs in the reduced MD data set have a PCC of more than 0.85. In other words, all correlations are now less than 0.85.

Now that a data set with reduced dimensionality has been created, the model that is to be used with this data set can now be explained. The total number of features that will be fed into the neural network will be 40, substantially less than the original 68. Therefore, the first layer of the neural network (the input layer) will have 40 neurons. The method proposed method by Heaton [17] for the number of neurons per hidden layer is employed here once gain. Thus, our only hidden layer will consists of 27 neurons. The output layer is simply one neuron. The model is trained for 500 epochs, however, for this model the L1 loss function as the criterion. The L1 loss function is otherwise known as the mean absolute error loss function (MAE). The L1 loss function is defined as follows:

$$MAE = \frac{1}{n}\sum_{j=1}^{n} |y_i - \hat{y}_j|$$

The advantage of using the L1 loss function is that we are now using linear criterion that does not penalize the model too harshly on outliers, thus this will create a model that makes more generic predictions as opposed to over fitted ones.After training the model over 500 epochs the model converged to a validation loss of 9.60 and a training loss of 10.33. The ReLu activation function is once again used for the hidden layer of this neural network.

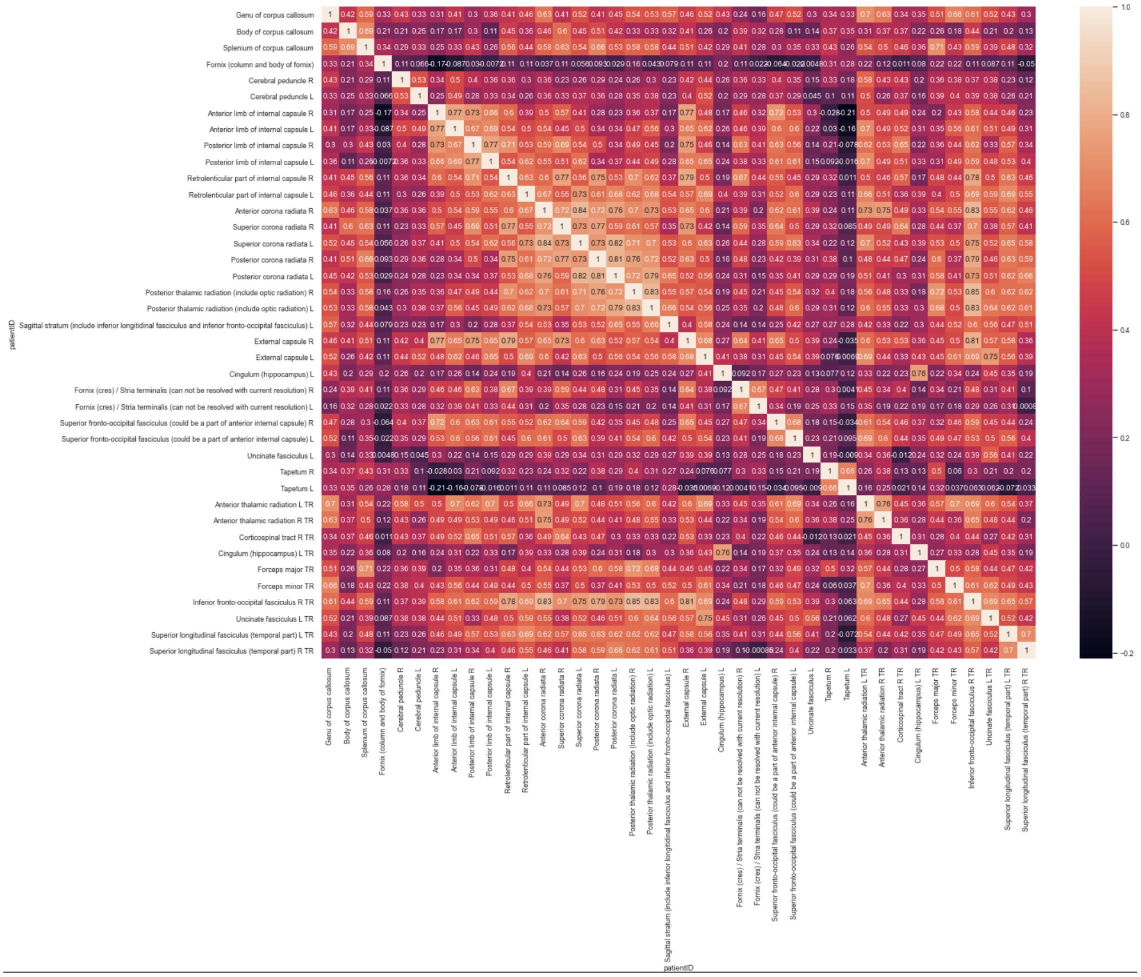


Figure 4.14: Correlation matrix after dropping 28 features.

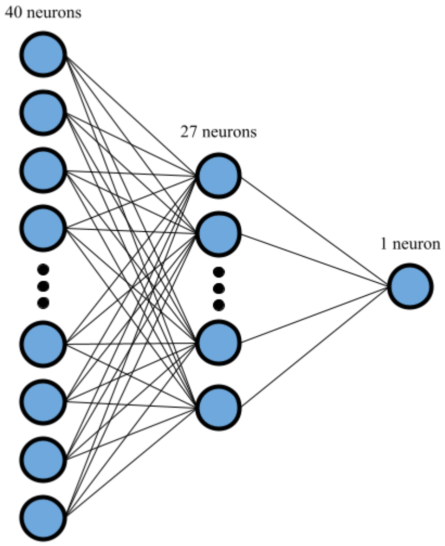


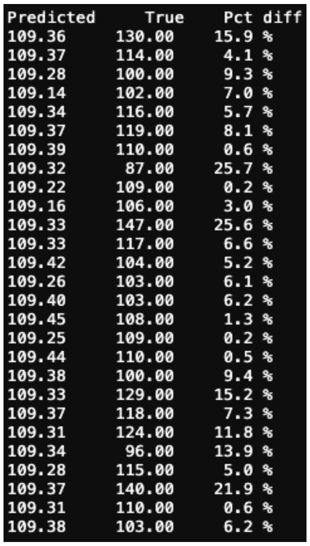
Figure 4.15: Model to be used after employing correlation matrix feature reduction technique.

4.3.1 Evaluating the Results

As stated previously, the model was trained over 500 epochs. Figure 4.16 gives an overview of the results. Again, it is clear that the model has over fitted from the data once again. The model is once again only predicting the average IQ score. There is some slight variation in the decimal part of the predictions, however, it is not nearly enough and a better model can surely be constructed.

If there is one thing that is obvious from the models that have been created so far is that they are all over fitting, and figure 4.17 confirms this claim by clearly showing that the true IQ scores range from just below 90 to below 150. However, the predicted IQ scores can be summarized as 109. Although the average absolute percent difference for the predictions is 8.24%, this is not a good predictive model as we know the predictions are simply the average IQ score.

The data set that is being used is small and the over fitting can be primarily attributed to the size of the data set, but there must be a different approach that



```
Predicted      True     Pct diff
109.36      130.00     15.9 %
109.37      114.00      4.1 %
109.28      100.00      9.3 %
109.14      102.00      7.0 %
109.34      116.00      5.7 %
109.37      119.00      8.1 %
109.39      110.00      0.6 %
109.32       87.00     25.7 %
109.22      109.00      0.2 %
109.16      106.00      3.0 %
109.33      147.00     25.6 %
109.33      117.00      6.6 %
109.42      104.00      5.2 %
109.26      103.00      6.1 %
109.40      103.00      6.2 %
109.45      108.00      1.3 %
109.25      109.00      0.2 %
109.44      110.00      0.5 %
109.38      100.00      9.4 %
109.33      129.00     15.2 %
109.37      118.00      7.3 %
109.31      124.00     11.8 %
109.34       96.00     13.9 %
109.28      115.00      5.0 %
109.37      140.00     21.9 %
109.31      110.00      0.6 %
109.38      103.00      6.2 %
```

Figure 4.16: Predictions for NN and using a correlation matrix for feature reduction.

yields better results. Given that we know that the data is normally distributed, instead another approach is to lever this property and use it to our advantage. A good approach is the the Z-score.

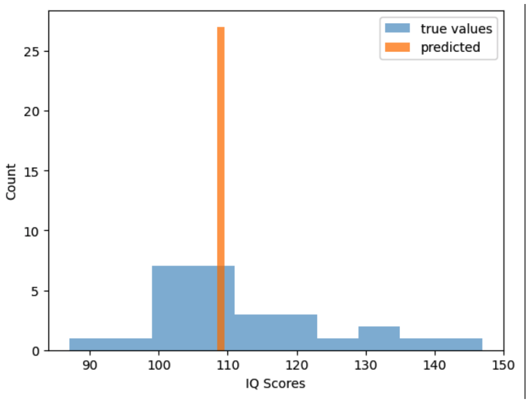


Figure 4.17: Distribution of predicted and true IQ scores.

4.4 NN, Correlation Matrix and Standardization

Up until now, the models that have been developed have yielded over fitted predictions that represent the average IQ score of the training data. That is a problem as we are not interested in 'predicting' the average IQ score for all healthy 8-year old children. This would be a simple calculation otherwise. It is true we are dealing with a small data set with high dimensionality, but one factor that has yet to be considered is the data itself. Because we are dealing with MD data, the data values themselves are very small.

Patient	Feature 1	Feature 2	Feature 3
patient x	0.000384	0.000507	0.000741

Table 4.1: Abstraction of the data, showing 3 of 68 features for some patient x

Table 4.1 gives an overview of 3 of the 68 features for some random patient in the MD data set. It is clear that the features the model is learning from are relatively small in terms of scale. This could be another factor that is negatively influencing the generalization of the predictions. Therefore, scaling the data to values that are described in terms of the population mean and population standard deviation should yield better results as the scaled values will now contain 'new' information. However, since we do not have the population mean or population standard deviation for the type of data we are working with, we will use sample means and sample standard deviation. Therefore, in this section the methodology will consist of using using the Z-score to standardize the data and then a correlation matrix will be used for feature reduction in an attempt to reduce the dimensionality of the data.

Before diving into specifics, the Z-score shall be explained. The Z-score essentially describes how many standard deviations a given value lies above or below the population mean [27]. The Z-score is formally defined as follows:

μ = population mean

n = sample size

σ = population standard deviation

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x-\mu)^2}{n}}$$

$$\text{Z-score} = \frac{x-\mu}{\sigma}$$

Using the Z-score gives the added advantage that it has been shown that standardized data yields better results in neural networks [28]. The data will be standardized and then the same method as the section above, using correlation matrix for feature reduction, will be used.

Now that the standardization methodology and the reasoning behind it has been explained, the model itself can be analyzed. After applying standardization to all 68 features of the data set, all feature pairs that had a correlation coefficient greater than 0.70 were found and then one of those features was dropped. In total 398 feature pairs were found and 56 unique features were dropped. This resulted in a new data set with only 12 features. The input layer of the neural network will have 12 neurons for the 12 features of the data set. As previously stated, we will use the method proposed by by Heaton [17] for the number of neurons in the hidden layer, thus our model has 9 neurons for the hidden layer. The output layer will consist of only one neuron. The model will use he L1 loss function, the Adam optimizer and it will be trained over 900 epochs. Figure 4.18 gives an over view of the model to be used for this method.

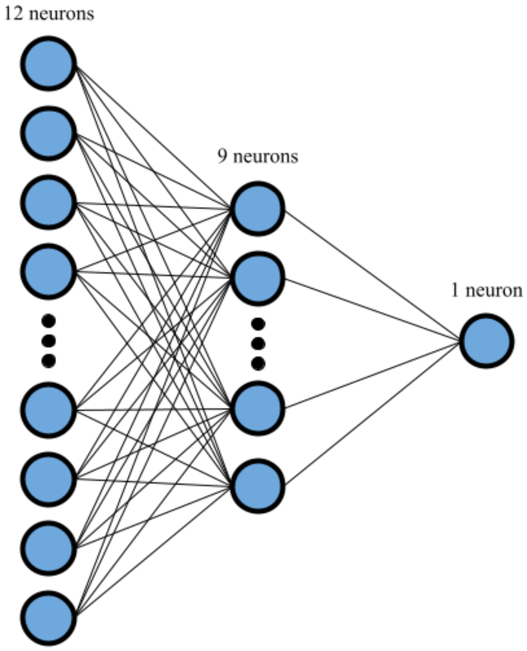
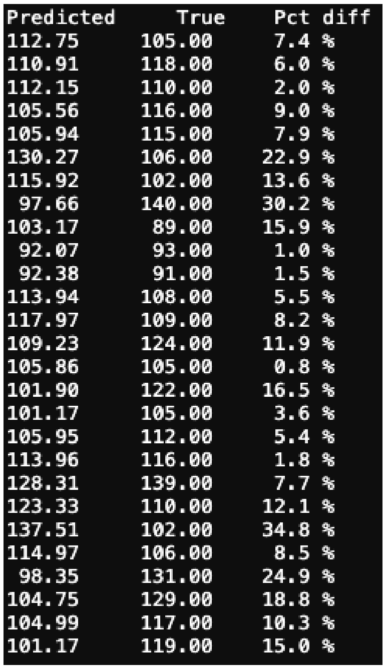


Figure 4.18: Model to be used after feature reduction and standardization

4.4.1 Evaluating the Results

As mentioned previously, the model was trained over 500 epochs. After training was complete, the training loss was 2.99 and the validation loss was 12.99.

It is clear from figure 4.19 that this model performs much better than he first 3 models. At first glance, there is no obvious over fitting. However, that does not mean that over fitting is non-existent in this model, it is just not as obvious as in the previous 3 models where the models predicted the average IQ of the training data set. This model yield better results and is learning valuable information from the training data. Although the predictions are not perfect, it can be observed that it does a good job of predicting a wide range of values with small errors. For example, the third IQ score prediction of 112.15 is very close to the true IQ score of 110 for that particular sample. Another observation that can be made is by looking at the 10th and 11th IQ score predictions of 92.07 and 92.38, respectively.



```
Predicted      True     Pct diff
112.75      105.00     7.4 %
110.91      118.00     6.0 %
112.15      110.00     2.0 %
105.56      116.00     9.0 %
105.94      115.00     7.9 %
130.27      106.00    22.9 %
115.92      102.00    13.6 %
 97.66      140.00    30.2 %
103.17       89.00    15.9 %
 92.07       93.00     1.0 %
 92.38       91.00     1.5 %
113.94      108.00     5.5 %
117.97      109.00     8.2 %
109.23      124.00    11.9 %
105.86      105.00     0.8 %
101.90      122.00    16.5 %
101.17      105.00     3.6 %
105.95      112.00     5.4 %
113.96      116.00     1.8 %
128.31      139.00     7.7 %
123.33      110.00    12.1 %
137.51      102.00    34.8 %
114.97      106.00     8.5 %
 98.35      131.00    24.9 %
104.75      129.00    18.8 %
104.99      117.00    10.3 %
101.17      119.00    15.0 %
```

Figure 4.19: Predicted and true IQ scores for this method.

Their respective true IQ scores are 93 and 91. Again, the error between the predicted IQ score and the true IQ score is fairly low.

Just as there are IQ score predictions that highlight the model's ability to generalize, there are also a few IQ score predictions that yield a large error. For example, observe the predicted and true IQ scores with a percent difference of 30.20. Here the predicted IQ score was 97.66 but the true IQ score was 140. These two are clearly on opposite sides of a distribution. This tells us that the either some important features were dropped that would have helped the model better estimate this IQ score, or that we simply need more data to create a more robust predictive model. It can be observed that there are a few other IQ scores that were predicted that have a large deviation from their respective true values.

Although these large deviations between predicted and true IQ scores exist, the

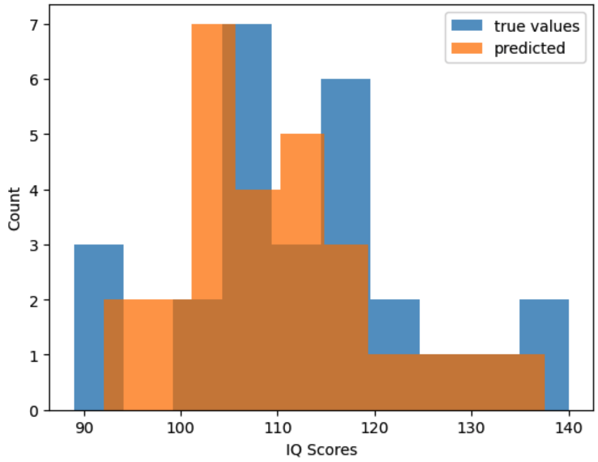


Figure 4.20: Distribution of predicted IQ scores and true IQ scores for this model

model performed relatively well. The average absolute percent difference in predicted IQ scores was 11.23%. Figure 4.20 gives an overview of the distribution between the predicted IQ scores and the true IQ scores. As opposed to the first 3 models, it is visibly clear that this model is able to generalize IQ scores a lot better and the range of the predicted IQ scores have a similar range to that of the true IQ scores.

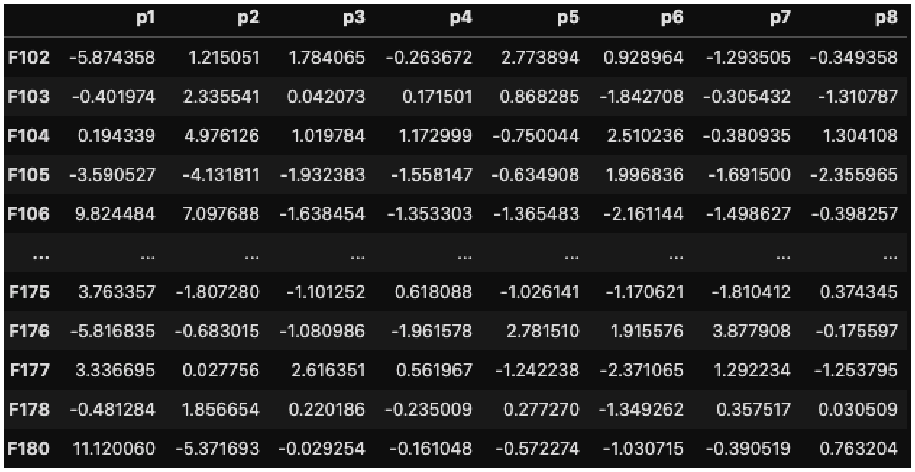
4.5 NN, PCA and Standardization

In the previous section, it was stated that the model was fed the standardized features performed much better than the first 3 approaches that did not make use of standardization. The previous approach also makes use of a correlation matrix for feature reduction. These two techniques led to the neural network being able to make better and more generalized predictions as opposed to the neural networks from the first 3 approaches. However, using a correlation matrix for feature reduction is more of a ’blind’ feature-reduction approach. This is because feature pairs above a specified correlation coefficient are identified, and then one of the features is dropped. However, this does not take into consideration what information is lost when these features are dropped from the data set. Therefore, in this section a new feature reduction technique is employed. For this approach, we will continue to standardize the features of the data set, but instead of using a correlation matrix as our feature reduction technique, the principal component analysis will instead be used.

The principal component analysis (PCA) is a technique that is often used to reduce the dimensionality of a data set [9]. The method for the PCA is somewhat complex, so it is recommended that the reader reference [9] and [29] for a deeper understanding of the PCA. However, when we apply the PCA as a feature reduction technique, it must be explained that features are not actually dropped and some are kept, instead what happens is that the features are used to transform that data into a new feature space where the 'new' features are actually principal components that each hold some variance (information) about the original features. Therefore, from now on, the 'new' features will be referred to as principal components as that is their appropriate name.

Researchers have used the PCA in to interpret high dimensional data in order in real world applications [29], and this has inspired the idea to use the PCA for feature reduction to analyze the MD data that has been used throughout this project. The PCA is such a powerful technique that researchers have used it as the basis for feature selection techniques. One of these techniques is the *Feature Selection for Classification using Principal Component Analysis and Information Gain* technique developed by Erick Odhiambo Omuya Et al [30].

Note that for this approach, the data will be first standardize using the methodology explained in the previous section. Once that standardization of each feature has been completed, the PCA will then be applied as the feature reduction technique. The PCA will yield 8 principal components that will represent our 'features', but remember that these are not directly related to the original features. Figure 4.21 gives an overview of some of the input data that will be used for the for the model, however note that this is before joining the input data with its corresponding label for the IQ scores. Notice that the ’features’ of the data set do not have specific names, instead the feature names are ’p1’, ’p2’ and so on. These ’features’ are the principal components that that resulted from using the PCA as a feature reduction technique.



	p1	p2	p3	p4	p5	p6	p7	p8
F102	-5.874358	1.215051	1.784065	-0.263672	2.773894	0.928964	-1.293505	-0.349358
F103	-0.401974	2.335541	0.042073	0.171501	0.868285	-1.842708	-0.305432	-1.310787
F104	0.194339	4.976126	1.019784	1.172999	-0.750044	2.510236	-0.380935	1.304108
F105	-3.590527	-4.131811	-1.932383	-1.558147	-0.634908	1.996836	-1.691500	-2.355965
F106	9.824484	7.097688	-1.638454	-1.353303	-1.365483	-2.161144	-1.498627	-0.398257
...	...	...	...	...	...	...	...	...
F175	3.763357	-1.807280	-1.101252	0.618088	-1.026141	-1.170621	-1.810412	0.374345
F176	-5.816835	-0.683015	-1.080986	-1.961578	2.781510	1.915576	3.877908	-0.175597
F177	3.336695	0.027756	2.616351	0.561967	-1.242238	-2.371065	1.292234	-1.253795
F178	-0.481284	1.856654	0.220186	-0.235009	0.277270	-1.349262	0.357517	0.030509
F180	11.120060	-5.371693	-0.029254	-0.161048	-0.572274	-1.030715	-0.390519	0.763204

Figure 4.21: An overview of some of the data samples and their respective principal components

The model will be fed these 8 principal components as the features, therefore, the input layer of the neural network will have 8 neurons. Again, we will use the proposed number of neurons by Heaton [17] for the hidden layer. Therefore, the hidden layer will contain 6 neurons. The output layer of the model will contain 1 neuron. The model will use the Adam optimizer, and the L1 loss function. The ReLu activation function will also be used for the hidden layer. The model will also be trained over 1,000 epochs. Figure 4.22 gives a general overview of the architecture of this model.

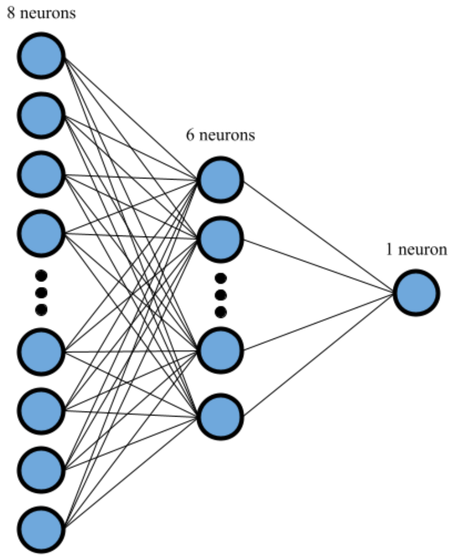
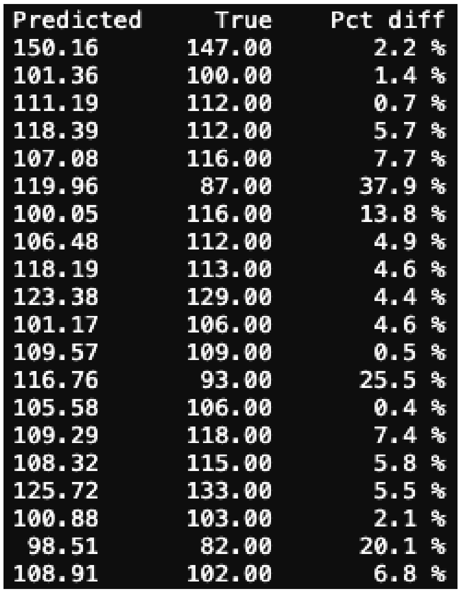


Figure 4.22: Model to be used after standardizing the data and employing the PCA as a feature reduction technique.

4.5.1 Evaluating the Results

After training the neural network over 1,000 epochs, the model converged to training MAE loss of 5.13 and a validation loss of 8.08.

Figure 4.23 shows the predicted IQ scores from the model and their respective true IQ scores. The figure also shows the percent difference between the predicted and the true IQ scores. At first glance it is once again clear that this model has not over fit the training data, unlike the first 3 models/methods that were used. This is good news once again because this demonstrates that the model was learned from the input data that was used. However, now the question remains, how well did it learn and how accurate are the predictions? Observe the first IQ score prediction of 150.16, its respective true IQ score is 147. Straight away it is impressive that the model was able to predict a score that seems to be above the 90th percentile of the IQ scores. This tells us that the standardizing the data and using the PCA for feature reduction both created information (standardization) and created principal



```
Predicted      True     Pct diff
150.16       147.00        2.2 %
101.36       100.00        1.4 %
111.19       112.00        0.7 %
118.39       112.00        5.7 %
107.08       116.00        7.7 %
119.96        87.00       37.9 %
100.05       116.00       13.8 %
106.48       112.00        4.9 %
118.19       113.00        4.6 %
123.38       129.00        4.4 %
101.17       106.00        4.6 %
109.57       109.00        0.5 %
116.76        93.00       25.5 %
105.58       106.00        0.4 %
109.29       118.00        7.4 %
108.32       115.00        5.8 %
125.72       133.00        5.5 %
100.88       103.00        2.1 %
 98.51        82.00       20.1 %
108.91       102.00        6.8 %
```

Figure 4.23: Predicted and true IQ scores for this model.

components that are useful to predicting high IQ scores. However, it is not enough to be able to predict high IQ scores, how does the model do with IQ scores in the lower percentiles? Observe the second to last IQ score prediction of 98.51 and its corresponding true IQ score of 82. It is once again impressive that the model was able to extract information from the principal components that allows it to make either high IQ score percentile predictions or lower percentile predictions. As a matter of fact, most IQ score predictions are actually very close to their corresponding true IQ scores. There are only 4 predictions where the percent difference between the predicted IQ score and the true IQ score is greater than 20%. Those predictions' percent differences are 37.90%, 13.80%, 25.50%, and 20.10%. All other predictions have a percent difference between the predicted IQ score and true IQ score of less than 7%. This is quite impressive given the amount of data that was available. The average absolute percent difference between the predicted IQ scores and true IQ scores is 8.09% for this model. This average lower that of the previous approach where the data was standardized and where a correlation matrix was used as the feature reduction technique.

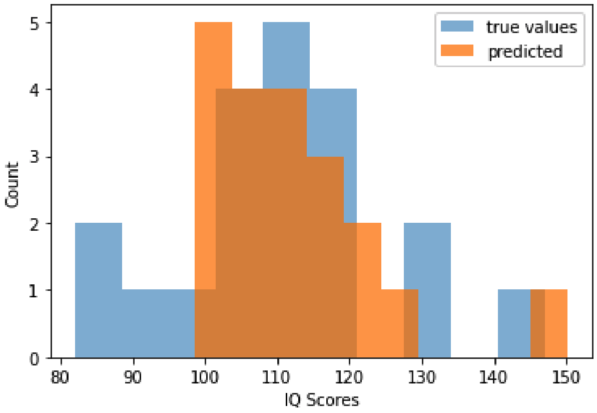


Figure 4.24: Distribution of predicted and true IQ scores.

Figure 4.24 gives an over view of the distribution of IQ score predictions and the true IQ scores. This figure allow one to visualize the breadth of the model when it comes to making predictions and it shows that the model is not over fitted as the predicted IQ scores have a decent range in relation to the true IQ scores. Overall, this model yields very well generalized predictions and it shows that standardizing the data and the PCA are methodologies that yield satisfactory IQ score predictions. It is possible that with even more data the model will only improve and make better IQ score predictions.

5 Conclusion

In this thesis, 5 different approaches for analyzing brain image data of healthy 8-year old children were discussed. The first approach was a vanilla neural network. The second method consisted of breaking the data into two smaller data sets, IQ scores above the 50th percentile and IQ scores below the 50th percentile. The third approach consisted of using a correlation matrix to try and reduce the number of features that were being red into the model in order to minimize the noise being fed into the model. The fourth approach was using a correlation matrix once again for feature reduction and standardizing the data. The last approach consisted of standardizing the data and using the PCA as our feature reduction technique. It was found that the first three methods were not good because they over fitted to the training data and were simply predicting the average IQ score of the training data.

Although the first three models yielded a low average percent difference between the predicted IQ scores and the true IQ scores, the predictions were useless and not insightful as they were simply an average with little to no deviation from the average IQ score. However, the fourth and fifth approaches yielded much more promising results. The fourth approach demonstrated that standardizing the data was a crucial step in making a more robust predictive model. The average absolute percent difference between the predicted IQ scores and the true IQ scores was 11.23%. Although this average absolute percent difference was higher than that of the first three approaches, it was a better predictive model as it was not over fitted to the training data and the predictions were more insightful. The fifth approach proved to be the best approach in predicting the IQ scores. For the fifth approach, the data was once again standardized and the PCA was used as the feature reduction technique. This approach led to much better predictions than the fourth approach and had an average absolute percent difference between the predicted IQ scores and true IQ scores of 8.90%. This is lower than the fourth model's average absolute percent difference of 11.23%. It was also demonstrated that most of the predictions of the fifth approach were very close to their true values, except for a few, but the large error in those predictions can be attributed to the small data set being used. A larger sample set would yield better results.

References

[1] T. Li, G. S. McCorkle, D. K. Williams, T. M. Badger, and X. Ou, "Cortical morphometry is associated with neuropsychological function in healthy 8-year-old children," 2020.

[2] Recommendations: Figuiring out how to bring unique joy to each member. [Online]. Available: https://research.netflix.com/research-area/recommendations

[3] B. Settless. How we learn how you learn. [Online]. Available: https://blog.duolingo.com/how-we-learn-how-you-learn/

[4] M. Stumpe, T. Lead, L. Peng, and P. Manager, "Assisting pathologists in detecting cancer with deep learning." [Online]. Available: https://ai.googleblog.com/2017/03/assisting-pathologists-in-detecting.html

[5] G. Forman and I. Cohen, "Learning from little: Comparison of classifiers given little training," 2004.

[6] N. A. Khovanova, T. Shaikhina, and K. K. Mallick, "Neural networks for analysis of trabecular bone in osteoarthritis," *Bioinspired, Biomimetic and Nanobiomaterials*, 2015.

[7] T. Shaikhina, D. Lowe, S. Daga, D. Briggs, R. Higgins, and N. Khovanova, "Machine learning for predictive modelling based on small data in biomedical engineering," *IFAC-PapersOnLine*, vol. 48, no. 20, 2015, 9th IFAC Symposium on Biological and Medical Systems BMS 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896315020765

[8] Z. M. Hira and D. F. Gillies, "A review of feature selection and feature extraction methods applied on microarray data," *Advances in Bioinformatics*, 2015. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4480804/

[9] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," 2016. [Online]. Available: http://dx.doi.org/10.1098/rsta.2015.0202

[10] D. Nandi, A. S. Ashour, S. Samanta, S. Chakraborty, M. A. Salem, and N. Dey, "Principal component analysis in medical image processing: a study," 2015. [Online]. Available: https://doi.org/10.1504/IJIM.2015.070024

[11] S. Zhang, "Studies of high spectral resolution atmospheric sounding data compression and noise reduction based on principal component analysis method," 2009.

[12] P. Wang, L. Li, and C. Yan, "Image classification by principal component analysis of multi-channel deep feature," 2017.

[13] L. Wang, C.-Y. Wee, H.-I. Suk, X. Tang, and D. Shen, "Mri-based intelligence quotient (iq)estimation with sparse learning," 2015.

[14] A. Mihalik, M. Brudfors, M. Robu, F. S.Ferreira, H. Lin, A. Rau, T. Wu, S. B.Blumberg, B. Kanber, M. Tariq, M. D. M. E. Garcia, C. Zor, D. I. Nikitichev, J. Mourao-Miranda, and N. P. Oxtoby, "Abcd neurocognitive prediction challenge2019: Predicting individual fluid intelligencescores from structural mri using probabilisticsegmentation and kernel ridge regression," 2019.

[15] K. Hilger, N. R. Winter, R. Leenings, J. Sassenhagen, T. Hahn, U. Basten, and C. J. Fiebach, "Predicting intelligence from brain gray matter volume," 2020.

[16] P. Denis Le Bihan MD, J. M. PhD, C. P. PhD, C. A. C. PhD, P. Sabina Pappata MD, N. M. MD, and H. C. MD, "Diffusion tensor imaging: Concepts and applications," 2001.

[17] J. Heaton, *Introduction to Neural Networks for Java, 2nd Edition*, 2nd ed. Heaton Research, Inc., 2008.

[18] A. Koutsoukas, X. L. Keith J. Monaghan, and J. Huan, "Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data."

[19] D. S. Mcleod. (2019) Introduction to the normal distribution (bell curve). [Online]. Available: https://www.simplypsychology.org/normal-distribution.html

[20] A. Viti, A. Terzi, and L. Bertolaccini, "A practical overview on probability distributions."

[21] Y. Perez-Riverol, M. Kuhn, J. A. Vizcaíno, M.-P. Hitz, and E. Audain, "Accurate and fast feature selection workflow for high-dimensional omics data," 2017.

[22] P. Schober, C. Boer, and L. A. Schwarte, "Correlation coefficients: Appropriate use and interpretation," *Anesthesia and Analgesia*, 2018.

[23] D. Wackerly, W. Mendenhall, and R. L. Scheaffer, *Mathematical Statistics with Applications 7th Edition*, 7th ed. Belmont, CA : Thomson Brooks/Cole, ©2008., 2008.

[24] T. Swinscow and M. J. Campbell, *Statistics at Square One*, 9th ed. Copyright BMJ Publishing Group, 1997.

[25] J. Rodgers and A. Nicewander, "Thirteen ways to look at the correlation coefficient," *The American Statistician*, 1988.

[26] M. Mukaka, "A guide to appropriate use of correlation coefficient in medical research," *Malawi Medical Journal*, 2012.

[27] H. Chubb and J. M. Simpson, "The use of z-scores in paediatric cardiology," *Annals of Pediatric Cardiology*, 2012.

[28] M.Shanker, M.Y.Hu, and M. Hung, "Effect of data standardization on neural network training," 1995.

[29] N. Salem and S. Hussein, "Data dimensional reduction and principal components analysis," 2019.

[30] E. O. Omuya, G. O. Okeyo, and M. W. Kimwele, "Feature selection for classification using principal component analysis and information gain," 2020.