Machine Learning: Genetic Disorder Prediction

Annalise Fry and Noah Houtchens

May 7, 2023

**Abstract**

Multiple genetic disorders can affect children and are becoming more common due to a “lack of understanding about the need for genetic testing” ¹. This project attempts to predict a potential genetic disorder using personal and medical information from past patients. To accomplish this, models like Elastic Net and ANN were trained using information from over 20,000 patients. Given the different models, our attained accuracy ranged from 32.15% using the Elastic Net model to 72.85% using the ANN model.

**Problem Definition and Goals**

The purpose of this project is to predict potential genetic disorders in children given different personal and medical information. The dataset was originally posted as a challenge on Hackerearth², and then on Kaggle in order to predict Genetic Disorders and Disorder Subclasses in an effort to bring understanding about the need for genetic testing during pregnancy. The goal of the project is to use different models to predict the potential disorders as accurately as possible. The dataset was obtained from Kaggle.com, which in turn was obtained from HackerEarth. A link to the dataset is provided in the R notebook for this project.

Before training and testing, the data needed to be cleaned and preprocessed, which is described in the Data Exploration and Preprocessing section of this document. Once prepared, multiple models were trained and tuned, of which the process is described in the Data Analysis and Results section. Lastly, the performances of the models and their accuracies were compared and can be found in the Conclusion section.

**Related Work**

Within the Kaggle listing for this dataset, one other notebook is posted. In this notebook, the disorder column was omitted from the dataset in order to compare the accuracy of prediction of with the column to without it. It was found in this notebook that the accuracy of the prediction fell from 87% to 55% without the column, as it is a highly correlated variable. Compared to our testing, the prediction accuracy varies by about 30%, dropping our ANN accuracy down to 41.30% and our Elastic Net down to a pitiful 16.53% which is only 5.53% better than random chance.

**Data Exploration and Preprocessing**

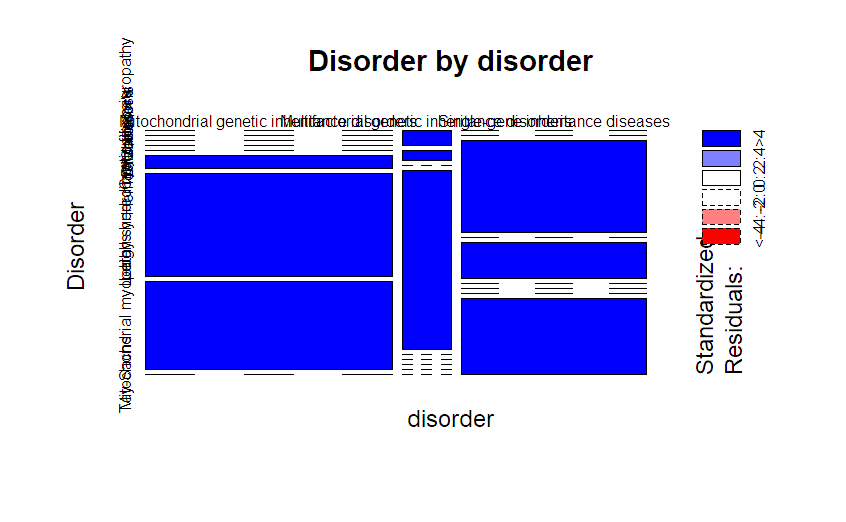
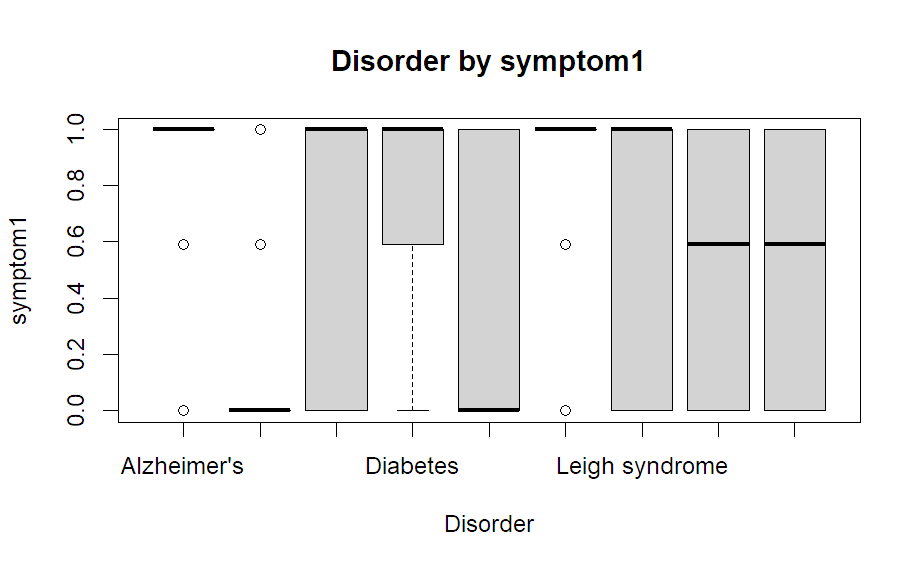
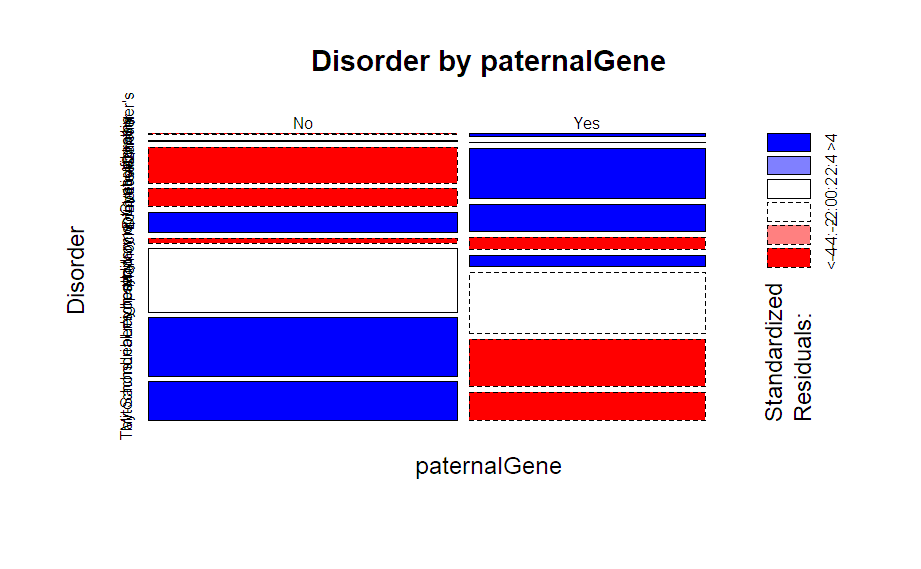
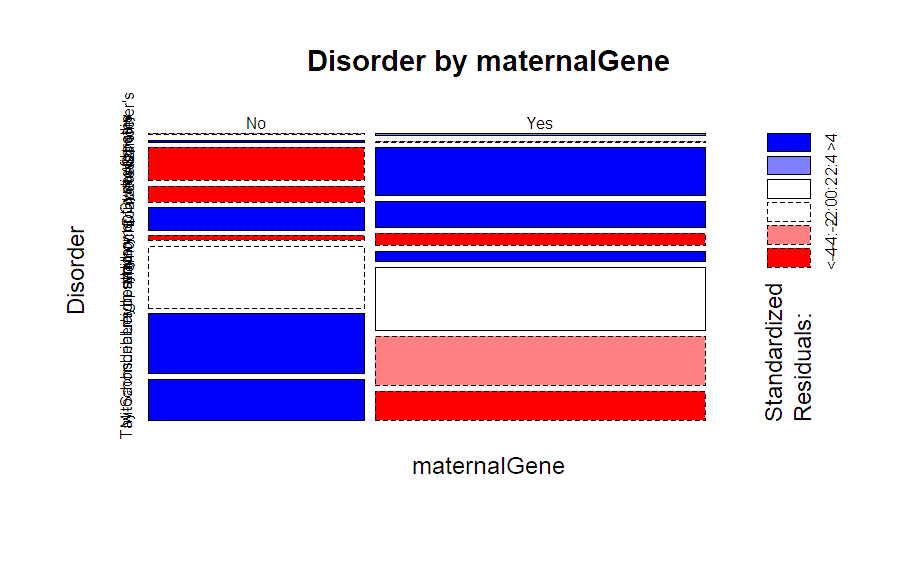
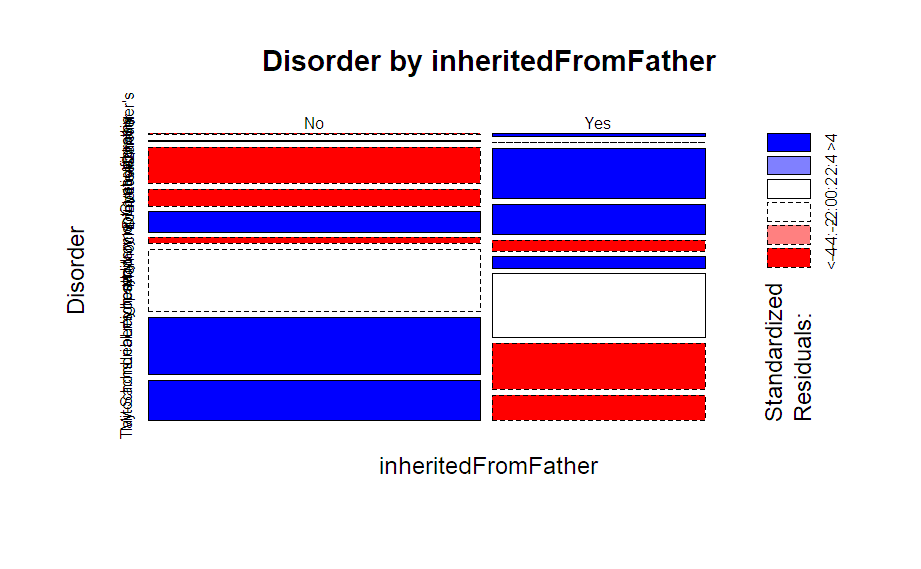
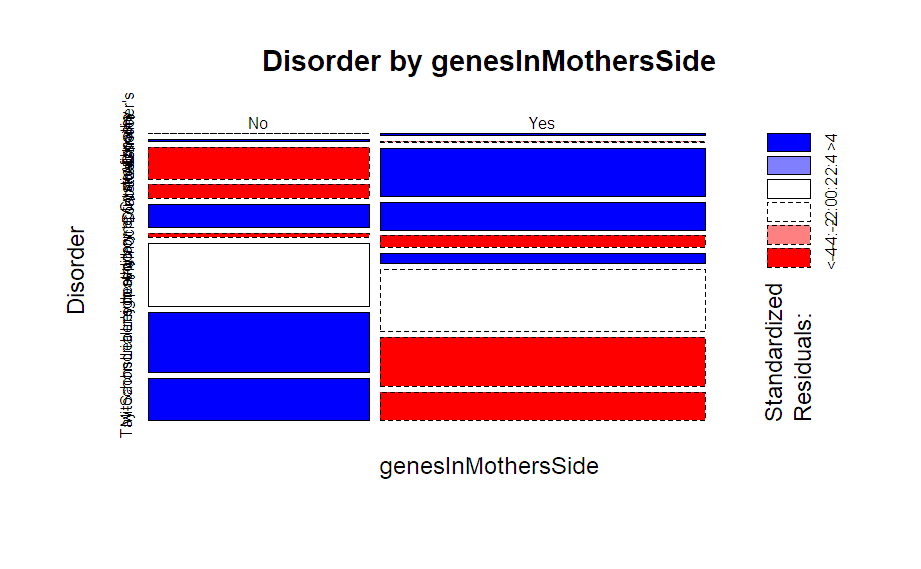
The initial dataset consisted of 22,083 observations (rows) and 45 variables (columns). The first step after importing the dataset was to rename the variables for clarity. Then, we removed variables that would not be necessary for modeling: patientID, patientFirstName, familyName, and fathersName. This left 22,083 observations and 41 variables, consisting of 25 categorical variables and 16 numerical variables:

patientAge: age of a patient - numerical  
inheritedFromMother: a gene defect in a patient’s mother - categorical  
inheritedFromFather: a gene defect in a patient’s father - categorical  
maternalGene: a gene defect in a patient’s maternal side of the family - categorical  
paternalGene: a gene defect in a patient’s paternal side of the family - categorical  
bloodCellCount: blood cell count of a patient - numerical  
mothersAge: age of a patient’s mother - numerical  
fathersAge: age of a patient’s father - numerical  
instituteName: medical institute where patient was born - categorical  
instituteLocation: location of medical institute - categorical  
status: whether a patient is deceased - categorical  
respiratoryRate: patient’s respiratory (breathing) rate - categorical  
heartRate: patient’s heart rate - categorical  
test1: one of a series of masked tests conducted on a patient - numerical  
test2: one of a series of masked tests conducted on a patient - numerical  
test3: one of a series of masked tests conducted on a patient - numerical  
test4: one of a series of masked tests conducted on a patient - numerical  
test5: one of a series of masked tests conducted on a patient - numerical  
parentalConsent: whether a patient’s parents approved treatment plan - categorical  
followUp: patient’s level of risk (how intense their condition is) - categorical  
gender: patient’s gender - categorical  
birthAphyxia: whether a patient suffered from birth asphyxia - categorical  
defectInAutopsy: whether a patient's autopsy showed any birth defects - categorical  
placeOfBirth: whether a patient was born in a medical institute or home - categorical  
folicAcidDetails: periconceptional folic acid - categorical  
seriousMaternalIllnessHistory: unexpected outcome of labor and delivery that resulted in significant short or long-term illness - categorical  
radiationExposureHistory: whether a patient has any radiation exposure history - categorical  
substanceAbuseHistory: whether a parent has a history of drug addiction - categorical  
assistedConception: type of treatment used for infertility - categorical  
anomalyHistory: whether mother had any anomalies in her previous pregnancies - categorical  
numberOfPriorAbortions: number of abortions that a mother had - numerical  
birthDefects: whether a patient has birth defects - categorical  
whiteBloodCellCount: patient's white blood cell count - numerical  
bloodTestResult: patient's blood test results - categorical   
symptom1: one of multiple masked symptoms a patient had - numerical  
symptom2: one of multiple masked symptoms a patient had - numerical  
symptom3: one of multiple masked symptoms a patient had - numerical  
symptom4: one of multiple masked symptoms a patient had - numerical  
symptom5: one of multiple masked symptoms a patient had - numerical  
disorder: overall genetic disorder that a patient has - categorical  
disorderSubclass: subclass of the disorder - categorical

The dataset had multiple missing values. We imputed any missing values of categorical data with mode of the data. For numerical data, we imputed any missing values with the mean of the data.

Additionally, there are several observations that are missing a value for the target variable, disorderSubclass. As there is no way to impute these observations without making the data biased, we removed these observations completely.

We were then left with 19,937 observations of 41 variables, with no missing values. Our data was ready to be explored via different tests in order to establish which variables are strongly associated with the target variable. For Categorical (disorderSubclass) to Categorical variables, we performed Mosaic Plots and Chi-Squared Tests and for Categorical (disorderSubclass) to Continuous variables, we performed Side-By-Side Boxplots and One-way ANOVA Tests. The following figures are the tests for some of the variables we ended up keeping for the data analysis.



Finally, we removed the variables from the dataset that were found to have low association with the target variable based on the p-values of the Chi-Squared tests and One-way ANOVA Tests at α = 0.01. We were left with 19,915 observations of 11 variables that we used for our models: genesInMothersSide, inheritedFromFather, maternalGene, paternalGene, bloodTestResult, symptom1, symptom2, symptom3, symptom4, symptom5, and disorder.

**Data Analysis and Results**

The models we used were an Artificial Neural Network, a Naïve Bayes classifier, a Logistic Model, an Elastic Net model, and a C5.0 Decision Tree.

**ANN:** Because this is a classification problem, the activation function of the ANN had to be SoftMax, and the loss function had to be sparse categorical cross-entropy. We one-hot-encoded the categorical data, which correlated to 20 input units. We used two hidden layers, two dropout layers, and an output layer with 9 outputs, one for each disorder subclass. The number of units in each hidden layer, dropout rate for each layer, learning rate, and batch size were all tuned using tfruns. The values we tested were as follows:

flags = list(

learning\_rate = c(0.1, 0.01, 0.001)

,batch\_size = c(32,16)

,units1= c(16,32,128)

,units2= c(16,32,128)

,dropout1= c(0.1, 0.2, 0.3)

,dropout2= c(0.1, 0.2, 0.3)

)

,sample = 0.01

This resulted in the following model:

Learning Rate: 0.001

units1: 128

units2: 128

batch size: 16

dropout1: 0.3

dropout2: 0.2

The number of epochs was chosen by fitting the subsequent model to the training data and finding the point where the model was not overfitting or underfitting. The number we decided on was 9.

Testing the ANN on the test data yielded an accuracy of 72.85%, which is excellent considering there were 9 output classes.

**Naïve Bayes:** The next model we trained was a Naïve Bayes classifier. This one was very straightforward and required no tuning of hyperparameters. In addition, the accuracy of the model was only slightly behind the ANN at 70.03%. Considering the severe reduction in time and energy compared to the ANN, the Naïve Bayes classifier was unquestionably more efficient in retrieving a quick, accurate answer.

**Logistic Model:** The next model we trained was a logistic regression model. For this one to work as a classifier, we rounded the output of the model to get an integer instead of a float and compared that to the test labels. Furthermore, the model only outputs values between 0-1, so the training data’s target variable had to be scaled down to that interval, then the output had to be scaled back up to the original interval. The accuracy of the model was not great at 36.80%, although still more than 3 times better than random guess.

**Elastic Net Model:** We trained another regression model in the form of the Elastic Net Model. This model also required we round the output to compare to the test labels. The accuracy of this model was the worst at 32.15%, but again, still ~3 times better than random guess. Both alpha and lambda were tuned using the tuneGrid with the following parameters:

tuneGrid = expand.grid(

alpha = seq(0, 1, length = 10),

lambda = 10^seq(-3, 3, length = 100)

)

The resulting values were:

alpha: 0.8888889

lambda: 0.003053856

**C5.0 Decision Tree:** The last model we created was the C5.0 Decision Tree. This model was also very straightforward and only required the number of trials to be tuned. We tried using a tuneGrid to autotune the number of trials, but it didn’t work, so we settled on running a loop from 1-30 to manually tune it based on validation data accuracy. The resulting number of trials were 26 with a validation accuracy of 71.99%. These parameters were then used to create a C5.0 Decision Tree using both the training and validation data as training data and produced an accuracy of 71.72% on the test data, placing it securely in 2nd place behind the ANN.

Each model’s performance was also measured using Error, Precision, and Recall for each class in the output. These results are available in the HTML and RMD files.

**Conclusion**

The ANN was unsurprisingly the best model at predicting the target variable, but the Naïve Bayes model was definitely shocking in terms of its accuracy, especially due to how fast it was compared to the ANN. The Logistic and Elastic Net models were also surprising in their abilities to generalize to a classification problem, even if they were subpar in their performances. The C5.0 also gave a stellar performance given its efficiency of implementation despite requiring some manual tweaking.

The biggest takeaway from this problem was that given a child with a known genetic disorder, a machine learning algorithm could easily be implemented to help determine with decent accuracy what that genetic disorder is. These models can definitely be improved given more time, more data (preferably with more statistically significantly associated variables), and more tuning.

Furthermore, it is apparent that there are other such problems specifically in the medical field that could greatly benefit from machine learning, especially when diagnosing a specific condition can be costly or timely—a machine learning model could help reduce the time and uncertainty of such diagnoses.

Another important takeaway is regarding the aforementioned capabilities of the Logistic and Elastic Net models. They clearly were not good picks for this particular problem but had the target variable been ordinal rather than nominal, we think the regression models could possibly perform on par with more sophisticated models such as ANNs.

**References**

[1] Kaggle, “Predict the Genetic Disorder,” [Online]. Available:  
 <https://www.kaggle.com/datasets/mukund23/predict-the-genetic-disorder?select=train.csv>

[2] Hackerearth, “Of Genomes and Genetics: HackerEarth Machine Learning Challenge,” [Online]. Available:  
 <https://www.hackerearth.com/challenges/new/competitive/hackerearth-machine-learning-challenge-genetic-testing/>