Group members

Name	Sec	BN
Habiba Mohamed Hanafy	1	25
Zeiad Ayman Mohammed	1	31
Omar Ayman Mohammed	2	2
Yara Hossam El-Din Mostafa	2	48

Content (Final Designing Blocks For Our Model)

- Importing libraries and loading data
- Data Wrangling (outliers and duplicates)
- Exploring and visualizing our dataset
- Dropping irrelevant features
- More preprocessing (Label encoding & Dummy encoding)
- Splitting the dataset into training and testing sets
- Fitting different models to the training set
- Testing different models and printing the results

Importing libraries and loading data

After importing the necessary libraries and models along with the metrics that we need to measure our performance, taking a look at our dataset reveals that some columns are completely irrelevant to our target. Said features can be dropped as a starter, such as “who completed the test” and “case number”.

Also, to understand our data better, we need to know what the first 10 columns correspond to, which are the answers to 10 questions asked to the caregiver of the child to fill in the screening test.

Variable in Dataset	**Corresponding Q-chat-10-Toddler Features**
A1	Does your child look at you when you call his/her name?
A2	How easy is it for you to get eye contact with your child?
A3	Does your child point to indicate that s/he wants something? (e.g. a toy that is out of reach)
A4	Does your child want to share interest with you? (e.g. pointing at an interesting sight)
A5	Does your child pretend? (e.g. care for dolls, talk on a toy phone)
A6	Does your child follow where you're looking?
A7	If you or someone else in the family is visibly upset, does your child show signs of wan9ng to comfort them? (e.g. stroking hair, hugging them)
A8	Would you describe your child's first words as:
A9	Does your child use simple gestures? (e.g. wave goodbye)
A10	Does your child stare at nothing with no apparent purpose?

Data Wrangling (checking for outliers and duplicates)

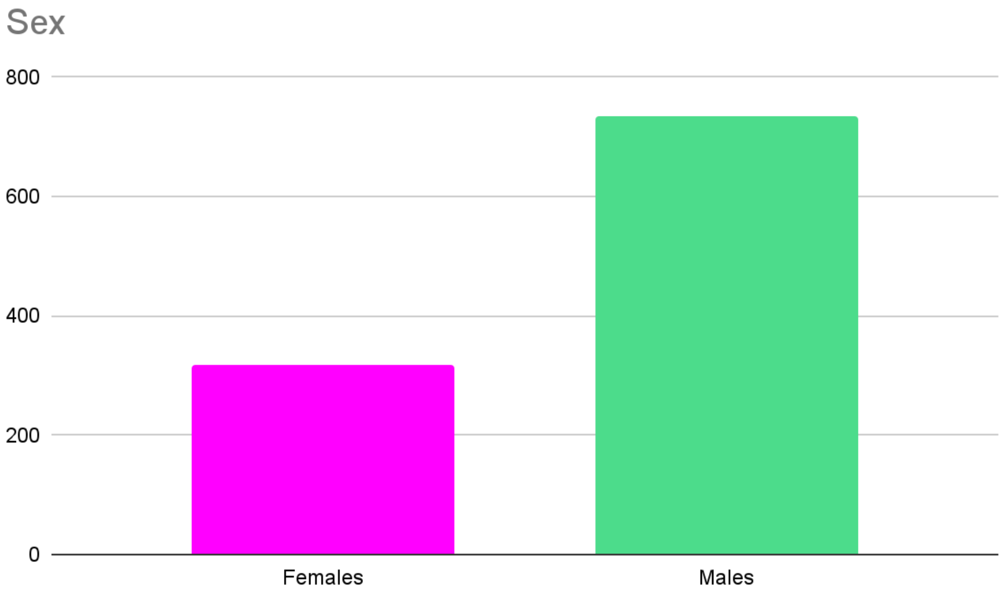
- To make sure the remaining columns don’t have outliers such as misspelling, we apply .unique() method on different categorical columns such as ”sex”, ”jaundice”, “Family_mem_with_ASD” ..etc. Fortunately, there’s no outliers in our dataset.
- We also check for duplicates using .duplicated().sum() and we luckily find none.

Data Exploration and visualization

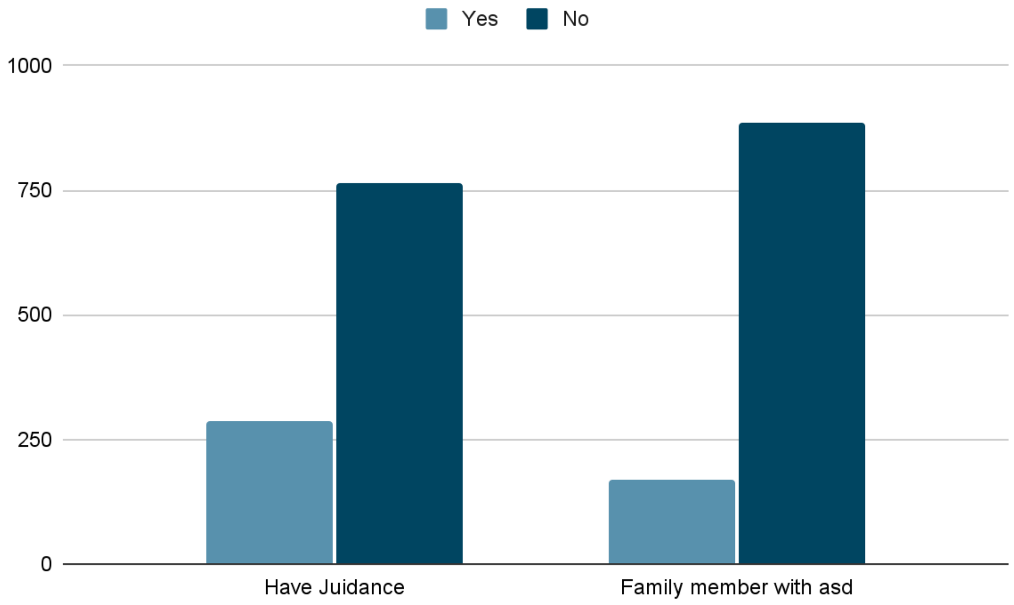
Now, we start exploring our features and measuring their relevance to our label. This is mainly to understand our data better and to make sure there is no bias in the dataset. We want to make sure that our data is representative of the population. This can be done using some visualizations.

First, we plot bar charts of **categorical features** we have such as: sex, jaundice, wheteher one of the parents have ASD.
We find out that:

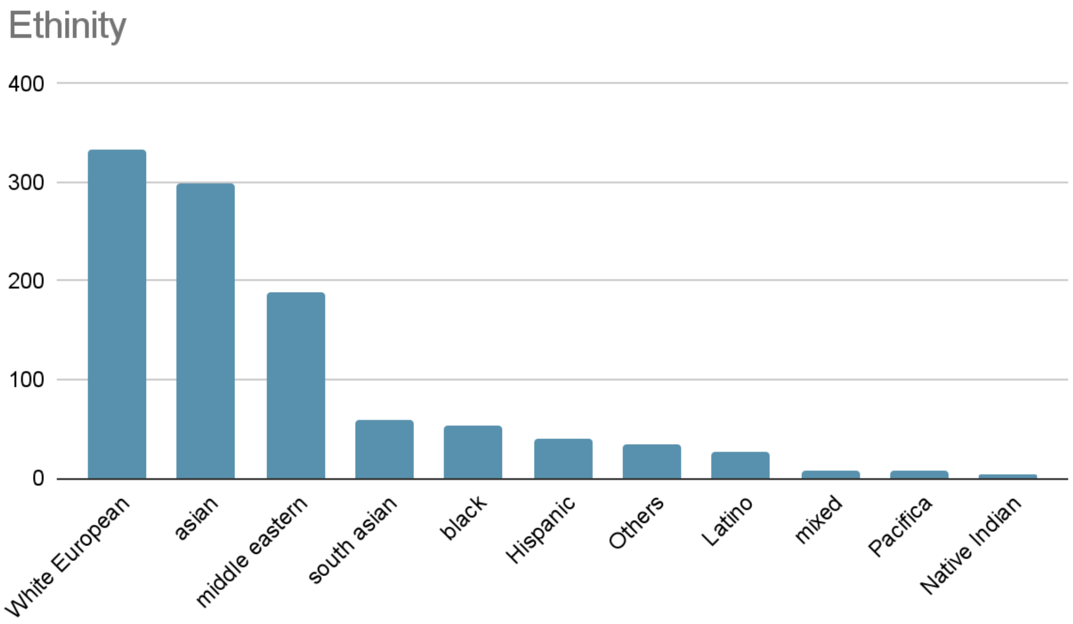
- For gender, our data has around 735 males and 319 females.



- While for jaundice, there are 288 of the records who were diagnosed and 766 healthy people.
- Also around 170 of the tested cases have a family member with ASD symptoms while approximately 884 don't.

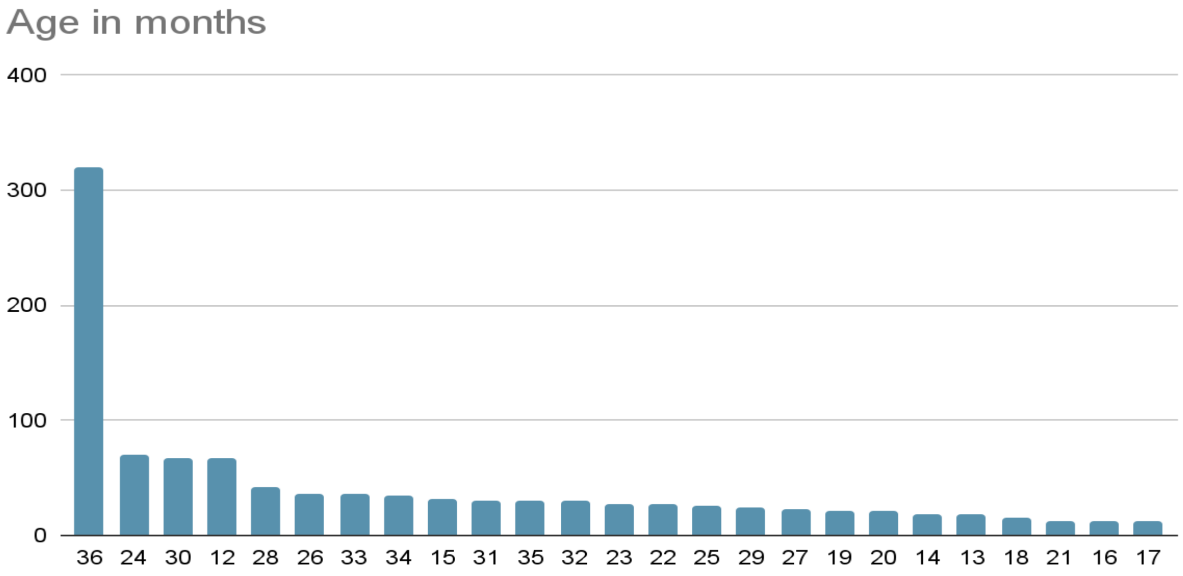


- For Ethnicity, the most prominent one is white european,followed by asians and middle easterns, with lowest count to native indians.



Next, for the **numerical features** (only age is numerical in our dataset), we plot a histogram to visualize the presence of different ages in our dataset.
Which results in the following observations:

- Most common age in our dataset is 36 months
- while the rarest is 17 months**Note**: dataset was collected with kids as the target audience. With the goal of predicting ASD in children.



Finally, we use a heatmap to plot the correlation between different columns in our dataset.
Our main focus is to identify the features with way too high correlation with other features that might be leaking info to the model (redundant) or may make the model biased.
We discover that:

- We notice a high correlation between the “Qchat-10-score” column and the first 10 columns which is a redundant feature (it sums the first 10 features) so we drop it to avoid overfitting.

Label encoding

We transfer our categorical features with string data into numbers. For example:

- Male:1, female:0
- yes:1, no:0

Dummy encoding

In order to represent ethnicity features in a proper manner, label encoding won't give us the best results so we resort to dummy encoding which is dividing ethnicity columns to a bunch of columns in a way to turn it into a binary feature. Like if we have for example 3 ethnicities white , black and asian we will have 3 columns(Features) where if the record is black will have 1 in the black column and 0 in white and asian.But one can see if we know that the record is not black and is not asian then it must be white.If we include all 3 columns we have fallen into dummy variable trap .So to avoid it we dropped the native indian ethnicity from the new columns that were made from ethnicity column.

Splitting the data

We split our data into training and testing sets with a test size of 0.3

Fitting our models

We try fitting our training set to multiple models such as:

- Logistic regression
- SVM
- Decision tree
- Naive Bayes
- KNN

Testing and printing the results

To measure the performance of our model, we don’t only check the accuracy, but it’s also crucial to check the confusion matrix. Especially the number of false negatives as we are tackling a medical problem.

- We can clearly see that logistic regression has the best result . It has 1 false negative and only 1 false positive. The score is also pretty good. (0.99)
- While other models didn’t perform as well,like SVM
- Clearly, the decision tree is not suitable for this problem since the dataset has many features and we know this can contribute to overfitting which decision tree is prone to. This is the case here as the score dropped from 1 for training to 0.908 for testing. It also has 10 false negatives which is alarming for a medical problem.
- For the knn the k selected was 10 and the accuracy is considered good , with false negative value of 7 , but still lags behind logistic regression and so did naive bayes .

The results

	Log Reg	SVM	Tree	NB	KNN
Confusion matrix	[94 1] [1 221]	[39 56] [1 221]	[85 10] [19 203]	[85 10] [10 212]	[90 5] [10 212]
Score	0.993	0.82	0.908	0.936	0.952