Water Predication Model

9

Features

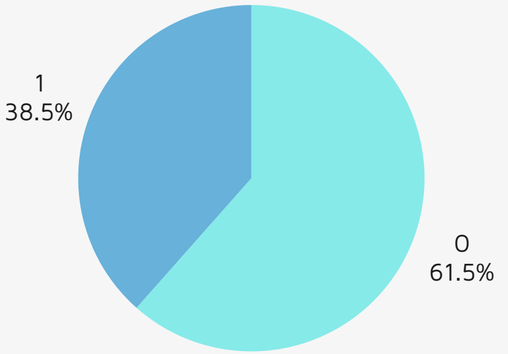
3276

entries

1

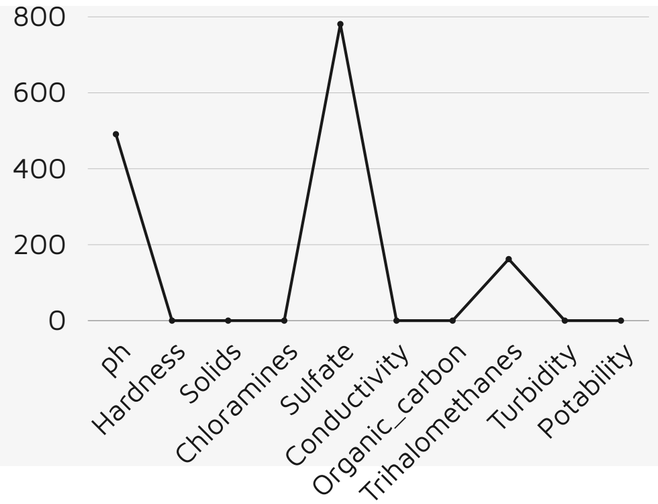
Target that is to see the potability of the water

DATA DISTRIBUTION



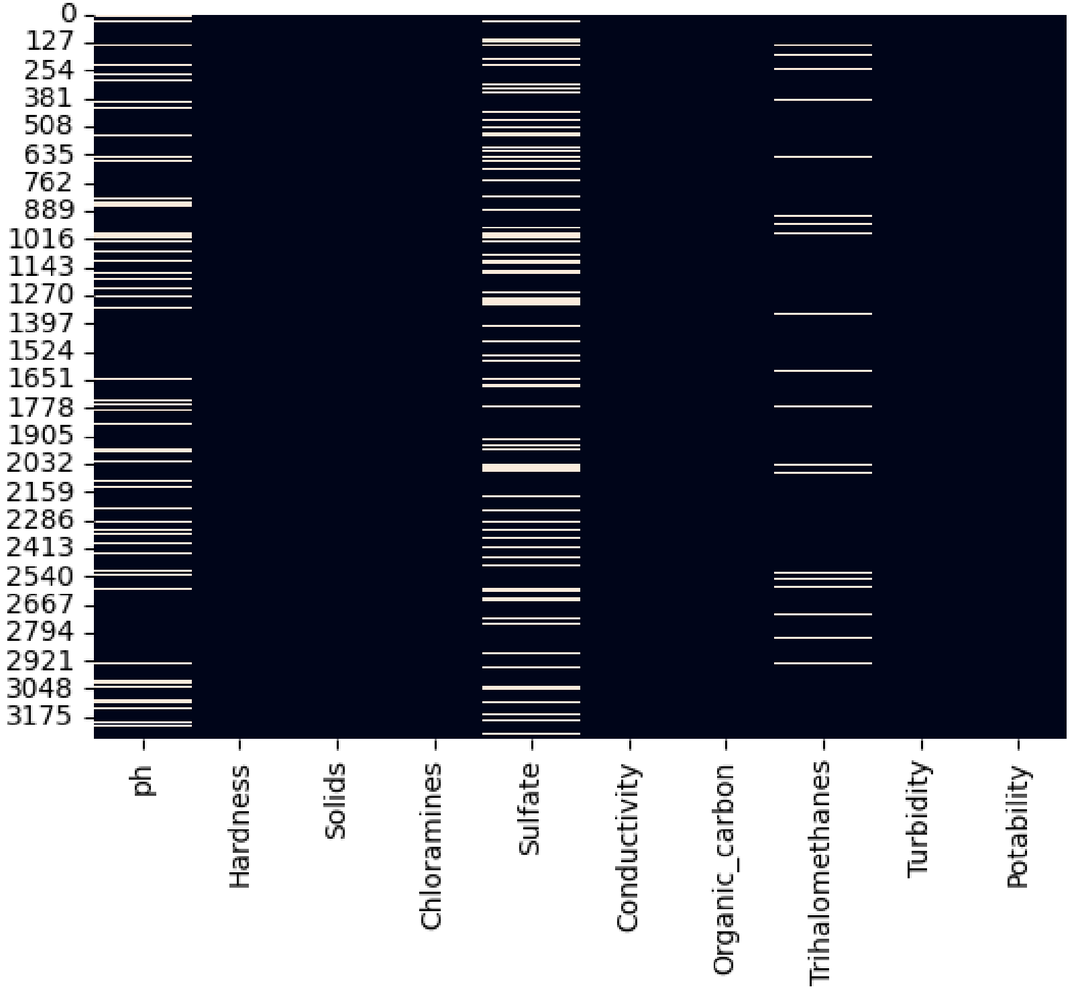
show a bar graph of the Potability column with 0 showing not portable and 1 showing portable

COLUMNS WITH MISSING DATA



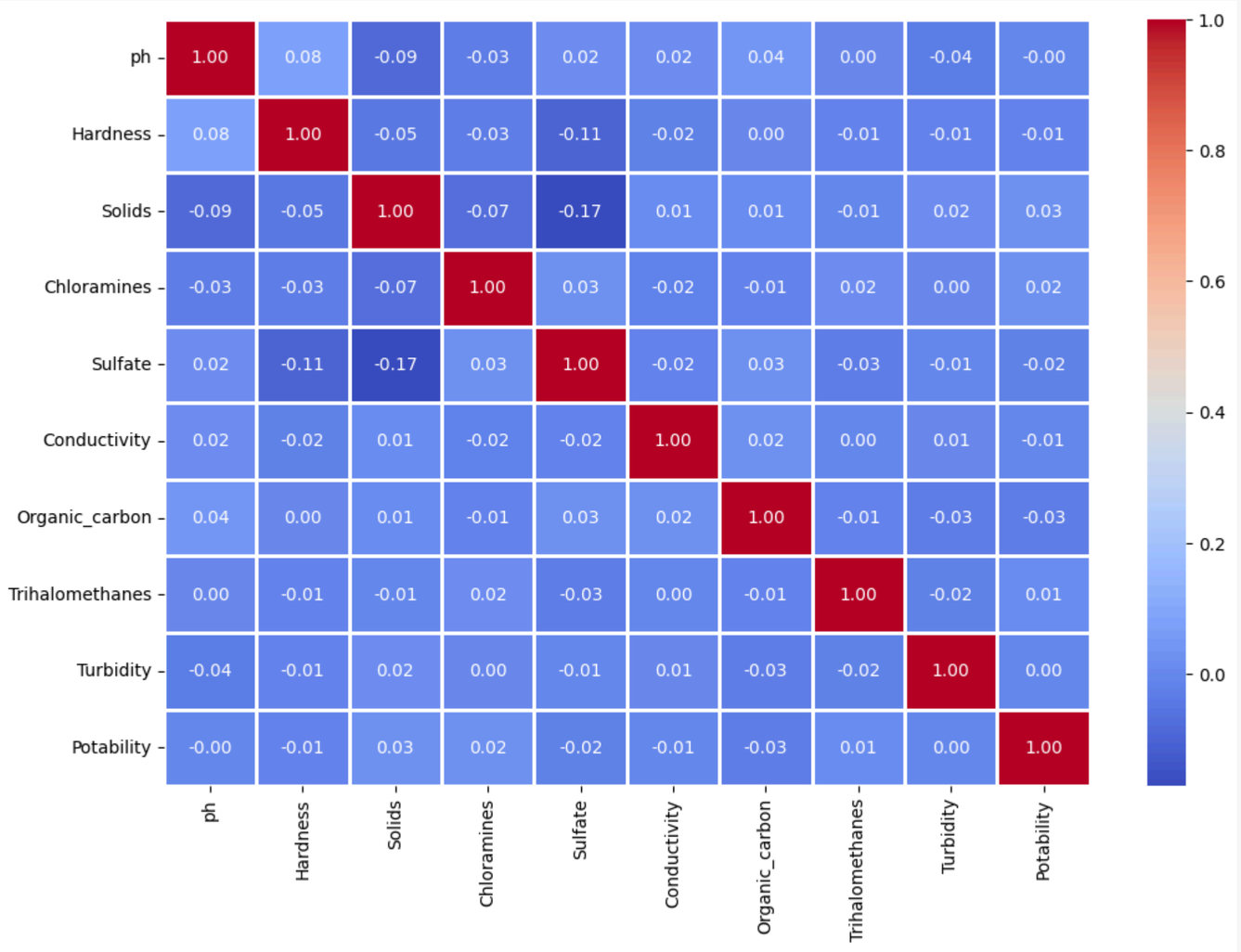
A line Graph showing the missing data in columns

STATISTICS ON DATA



This heatmap provides a concise overview of the missing data patterns, allowing us to identify features or observations that may require further investigation or imputation techniques

CORRELATION OF THE DATA



GETTING THE FEATURES AND TARGET

features

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity
0	7.080795	204.890455	20791.318981	7.300212	368.516441	564.308654	10.379783	86.990970	2.963135
1	3.716080	129.422921	18630.057858	6.635246	333.775777	592.885359	15.180013	56.329076	4.500656
2	8.099124	224.236259	19909.541732	9.275884	333.775777	418.606213	16.868637	66.420093	3.055934
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436524	100.341674	4.628771
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075

FEATURES

- 'ph
- Hardness
- Solids
- Chloramines
- Sulfate
- Conductivity
- Organic_carbon
- Trihalomethanes
- Turbidity'

TARGET

- Potability

BUILDING THE MODELS

1ST MODEL

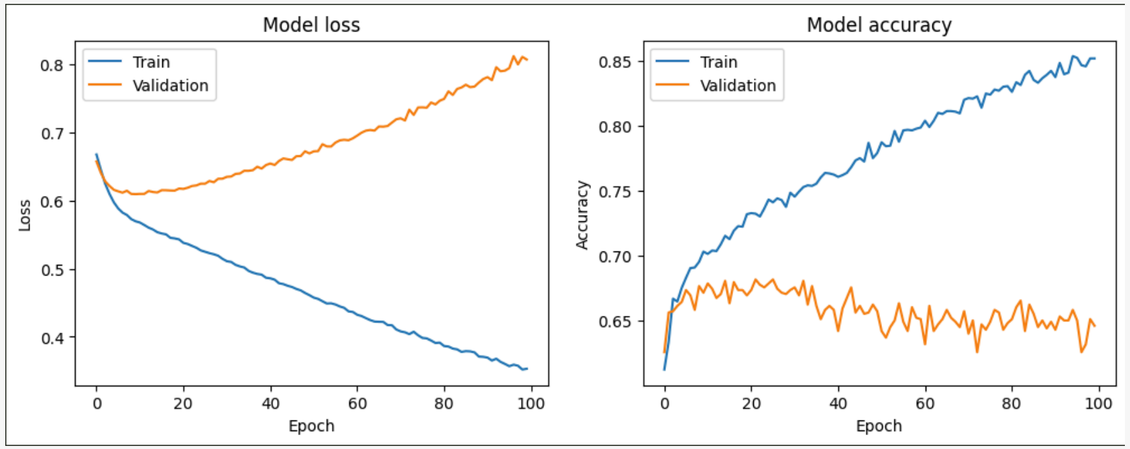
Without any early stopping

Without any L1 or L2,

MODEL: "SEQUENTIAL"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	640
dense_1 (Dense)	(None, 32)	2,080
dense_2 (Dense)	(None, 1)	33

TRAINING & VALIDATION ACCURACY VALUES



OBSERVATIONS

84%

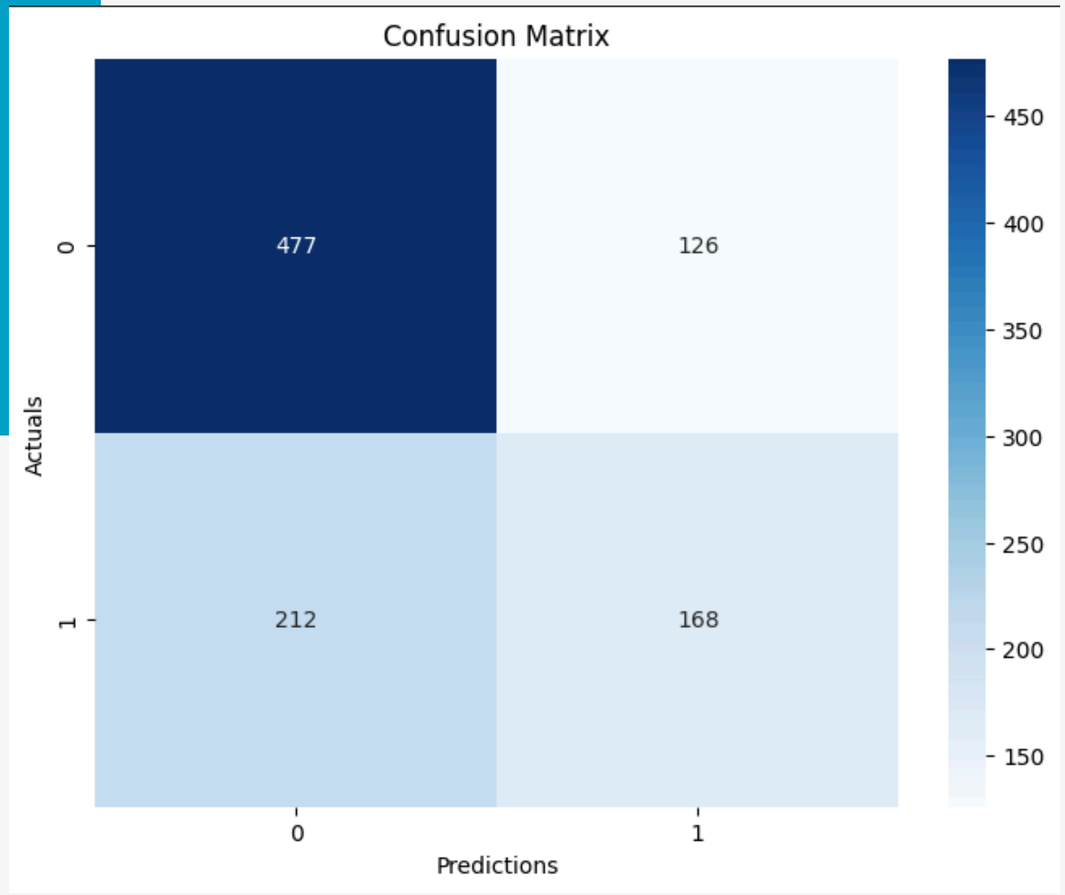
Training accuracy: 0.8466,

64%

When it came to Validation loss we had about 64%

Looking s at the results we had in this model, we noticed that our mode was overfitting when it come to the training datato

CONFUSION MATRIX



BUILDING THE MODELS

2ND MODEL

In the second model, we are interested in seeing the results of the model training and validation when we add the early stopping with l1 normalization

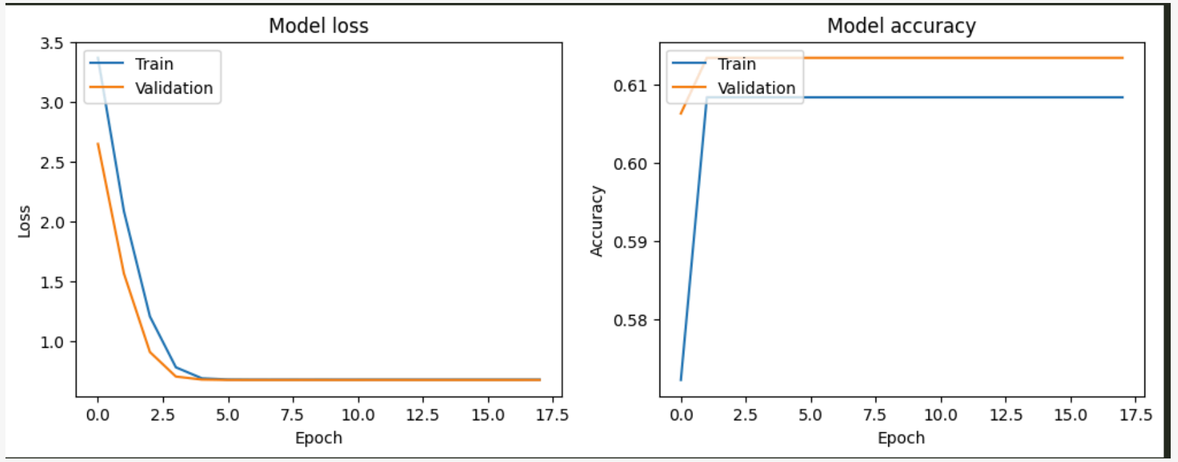
Early stopping

We are using L1 normalization with the learning rate of 0.01

MODEL: "SEQUENTIAL"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	640
dense_1 (Dense)	(None, 32)	2,080
dense_2 (Dense)	(None, 1)	33

TRAINING & VALIDATION ACCURACY VALUES



OBSERVATIONS

61%

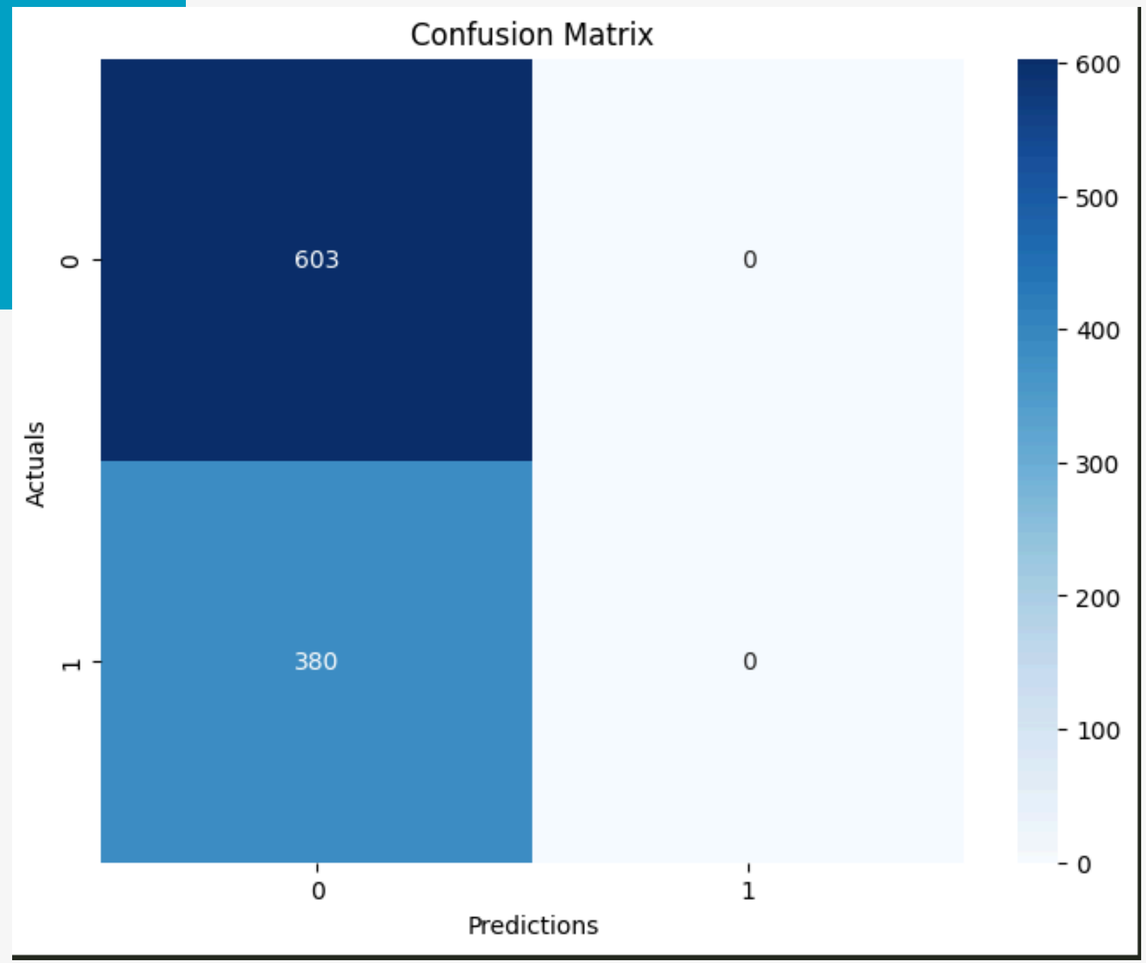
Training accuracy: 0.6156,

61%

When it came to Validation loss, we had about 61.34%

when we added the l1 norm with the learning rate of 0.01, we had a wonderful improvement with our model. we can clearly see an improvement

CONFUSION MATRIX



BUILDING THE MODELS

3RD MODEL

In the second model, we are interested in seeing the results of the model training and validation when we add the early stopping with l2 normalization

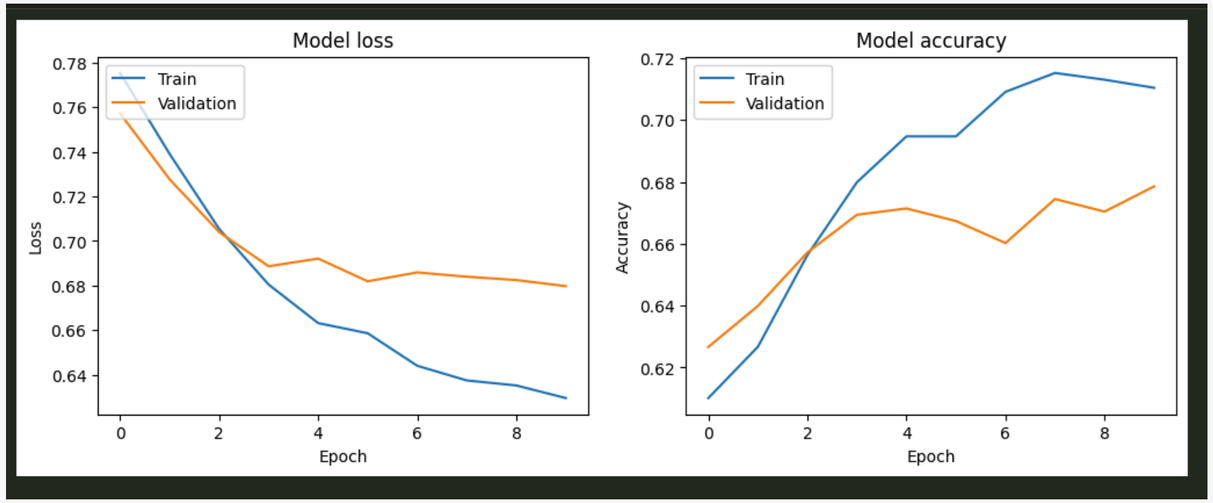
Early stopping

We are using L2 normalization with the learning rate of 0.01

MODEL: "SEQUENTIAL"

Layer (type)	Output Shape	Param #
dense_6 (Dense)	(None, 64)	640
dense_7 (Dense)	(None, 32)	2,080
dense_8 (Dense)	(None, 16)	528
dense_9 (Dense)	(None, 8)	136
dense_10 (Dense)	(None, 1)	9

TRAINING & VALIDATION ACCURACY VALUES



OBSERVATIONS

70%

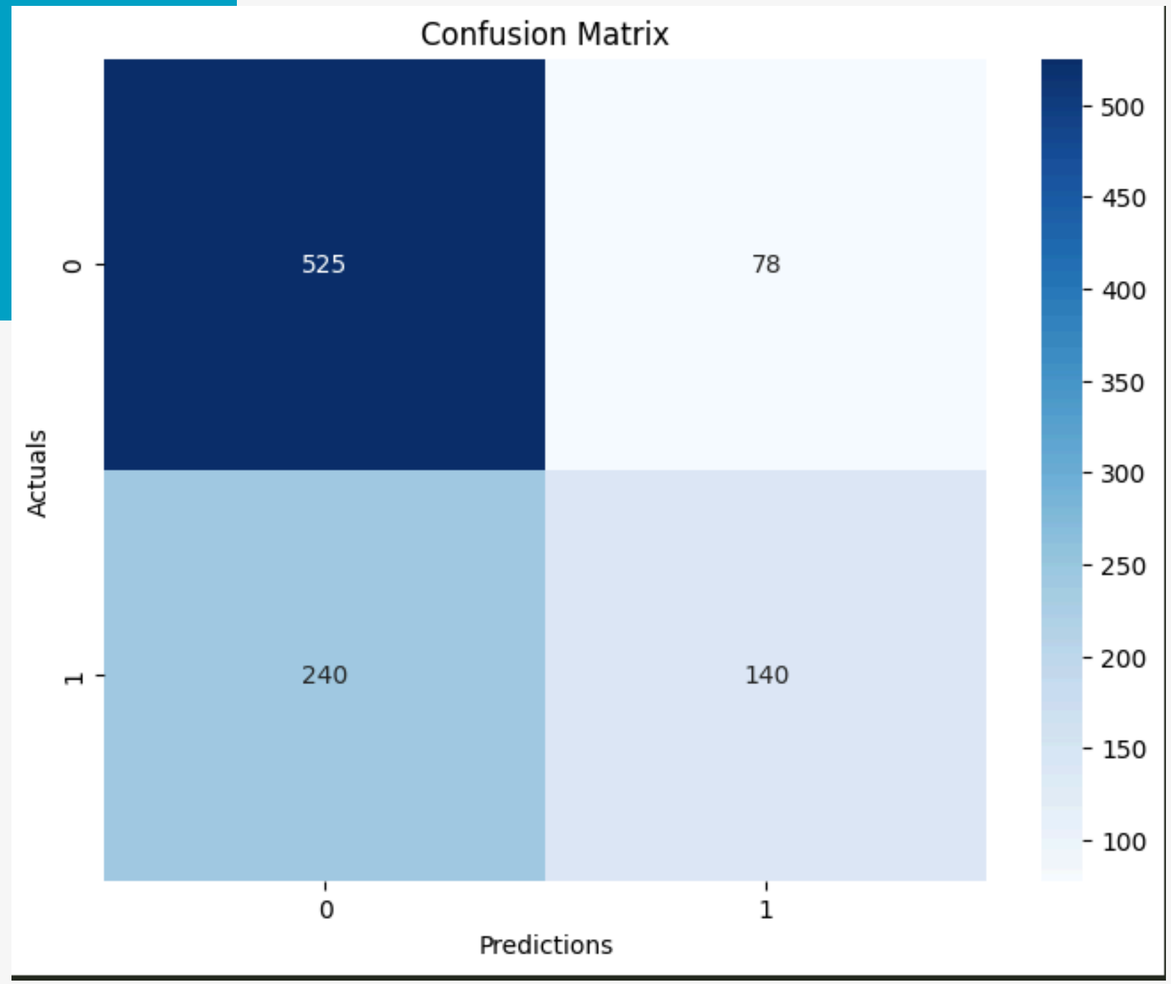
Training accuracy: 0.7067,

67%

When it came to Validation loss, we had about 67.85%

when we added the l1 norm with the learning rate of 0.01, we had a wonderful improvement with our model. we can clearly see an improvement

CONFUSION MATRIX



BUILDING THE MODELS

4TH MODEL

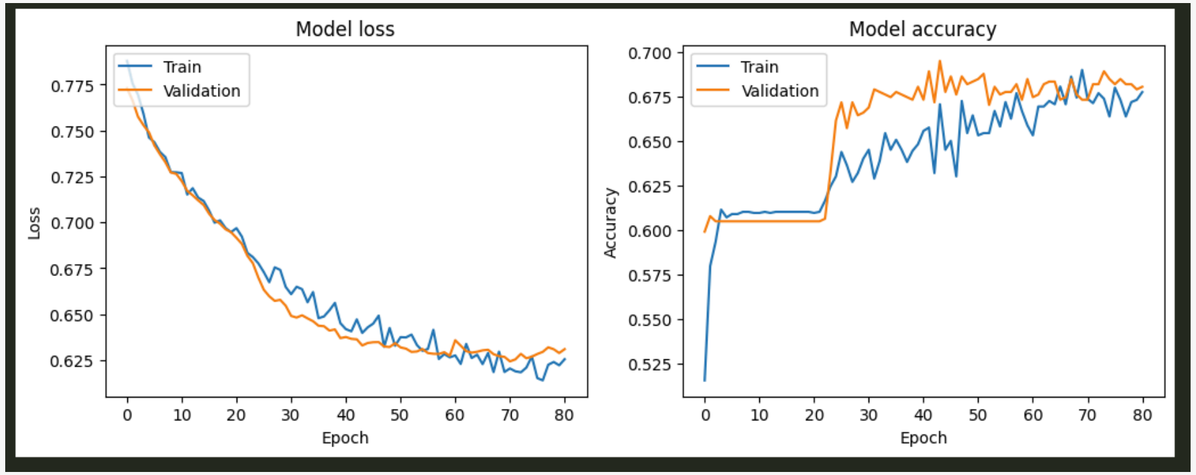
In the last model, we are sticking to using the L2 regularization and
We are also adding the dropout in our model,
We are also increasing the number of neurons in the model to make it learn more about the data.

Early stopping, More Neurons, also adding a dropout
We are using L2 normalization with the learning rate of 0.01

MODEL: "SEQUENTIAL"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	640
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2,080
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 16)	528
dropout_2 (Dropout)	(None, 16)	0
dense_3 (Dense)	(None, 8)	136
dropout_3 (Dropout)	(None, 8)	0
dense_4 (Dense)	(None, 1)	9

TRAINING & VALIDATION ACCURACY VALUES



OBSERVATIONS

69%

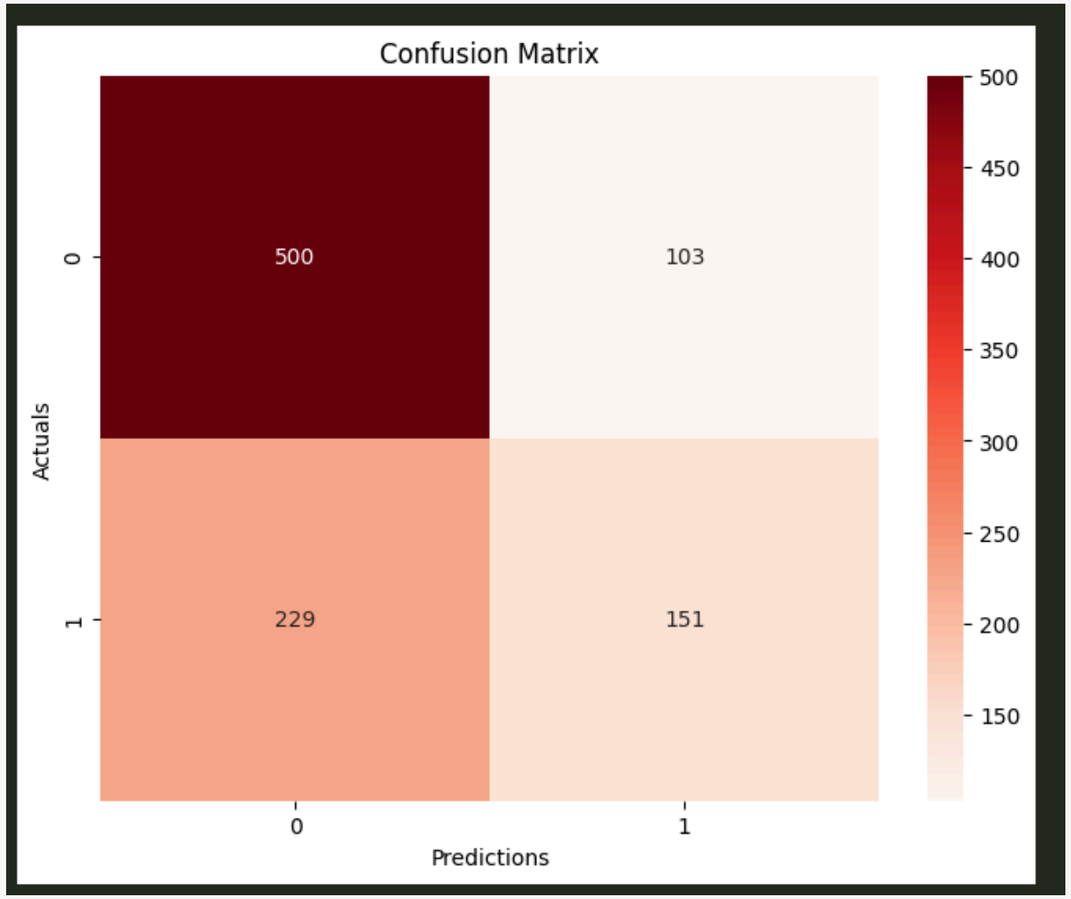
Training accuracy: 69,

63%

When it came to Validation loss, we had about 63%

Having this kind of model, we were able to fully map the training validation together with the validation. At the same time; we noticed an increase in accuracy with a decrease in loss..

CONFUSION MATRIX



BUILDING THE MODELS

5TH MODEL

In the last model, we are sticking to using the L2 regularization, and We are also adding the dropout in our model,
We are also increasing the number of neurons in the model to make it learn more about the data. However, we wanted to test using the Softmax in the last dense layer.

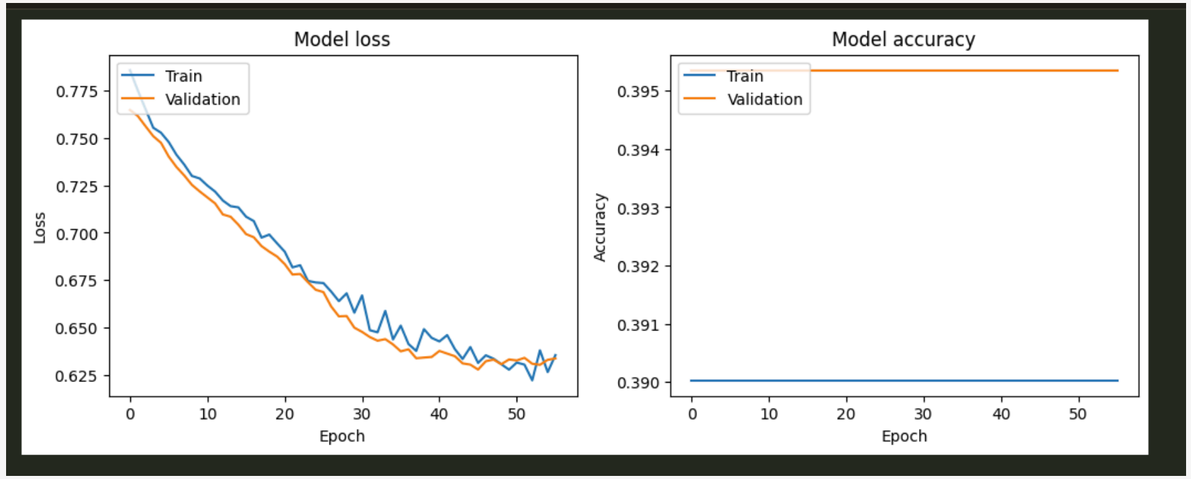
Early stopping, More Neurons, also adding a dropout

We are using L2 normalization with the learning rate of 0.01. We are also using softmax in the work.

ADDING SOFTMAX IN THE LAST LAYER

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	640
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2,080
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 16)	528
dropout_2 (Dropout)	(None, 16)	0
dense_3 (Dense)	(None, 8)	136
dropout_3 (Dropout)	(None, 8)	0
dense_4 (Dense)	(None, 1)	9

TRAINING & VALIDATION ACCURACY VALUES



OBSERVATIONS

37%

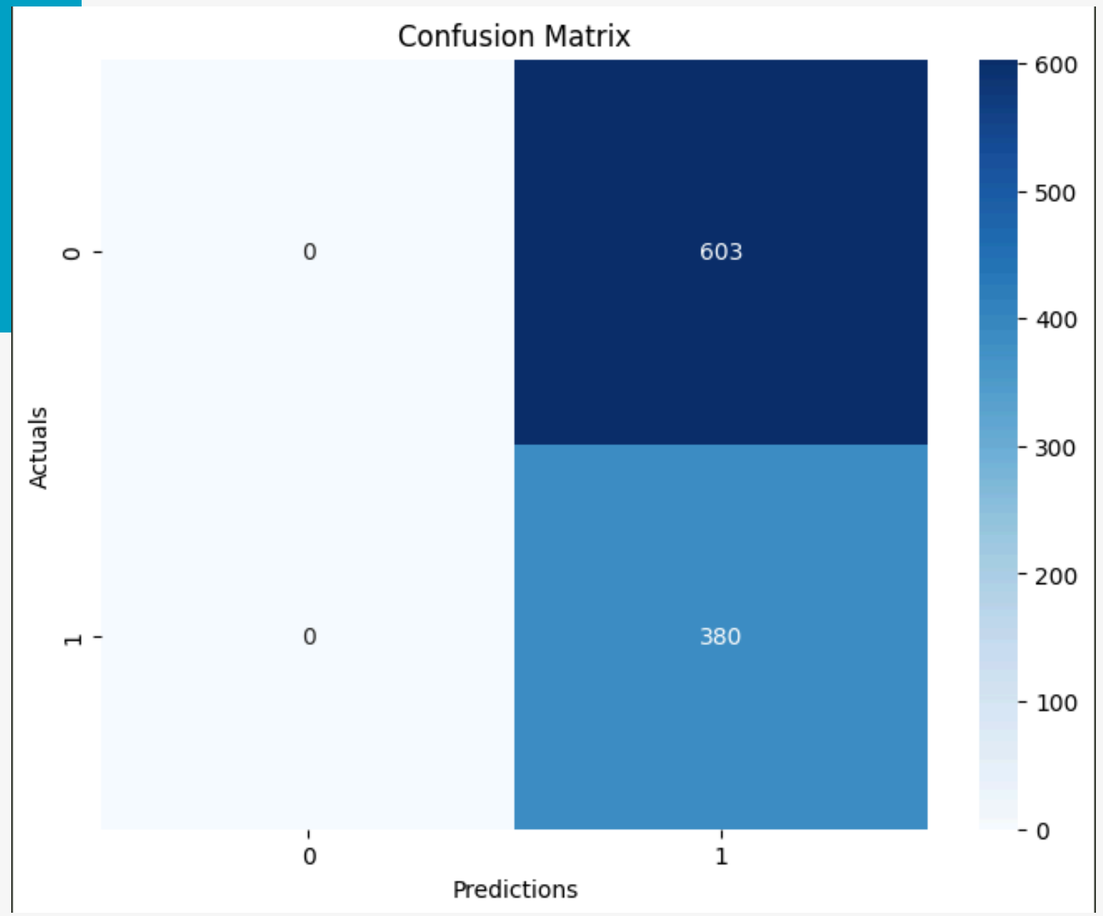
Training accuracy: 0.3704,

39%

When it came to Validation loss, we had about 39.53%

Adding g the softmax as an activation function in the last layer dramatically reduced the accuracy of our model by almost half percent

CONFUSION MATRIX



BUILDING THE MODELS

6TH MODEL

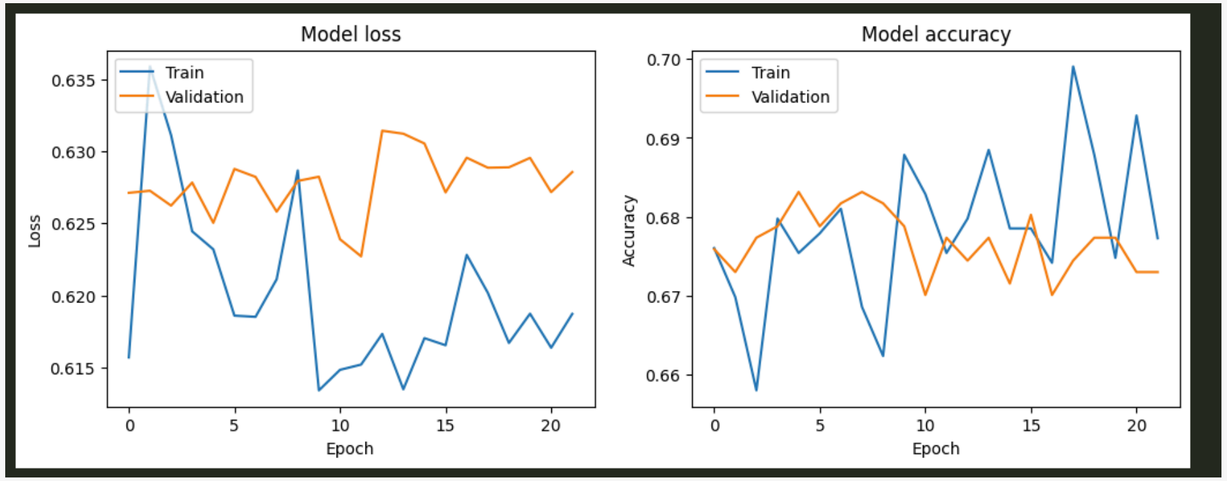
In the last model, we are sticking to using the L2 regularization, and We are also adding the dropout in our model,
We are also increasing the number of neurons in the model to make it learn more about the data. However, we wanted to test using the Softmax in the last dense layer.

Early stopping, More Neurons, also adding a dropout
We are using L2 normalization with a learning rate of 0.01. We are also using RMSpop optimiser in the work.

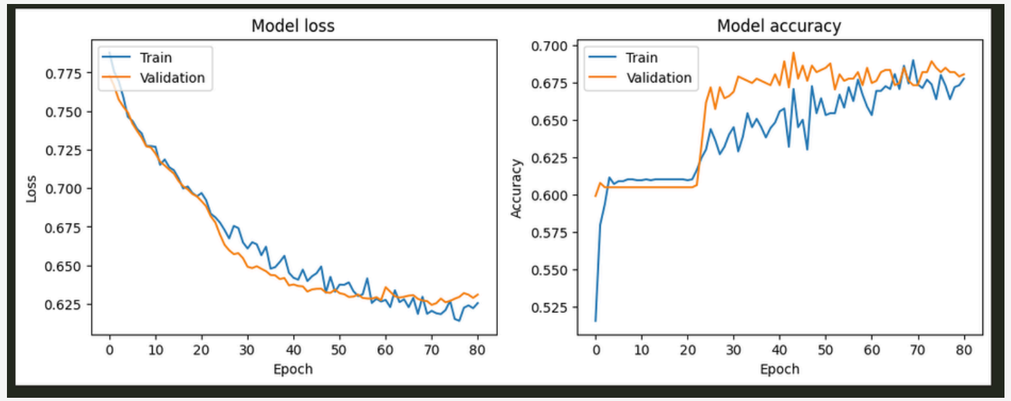
WE ALSO WANTED TO USE THE RMSPOP OPTIMISER

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	640
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2,080
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 16)	528
dropout_2 (Dropout)	(None, 16)	0
dense_3 (Dense)	(None, 8)	136
dropout_3 (Dropout)	(None, 8)	0
dense_4 (Dense)	(None, 1)	9

TRAINING & VALIDATION ACCURACY VALUES



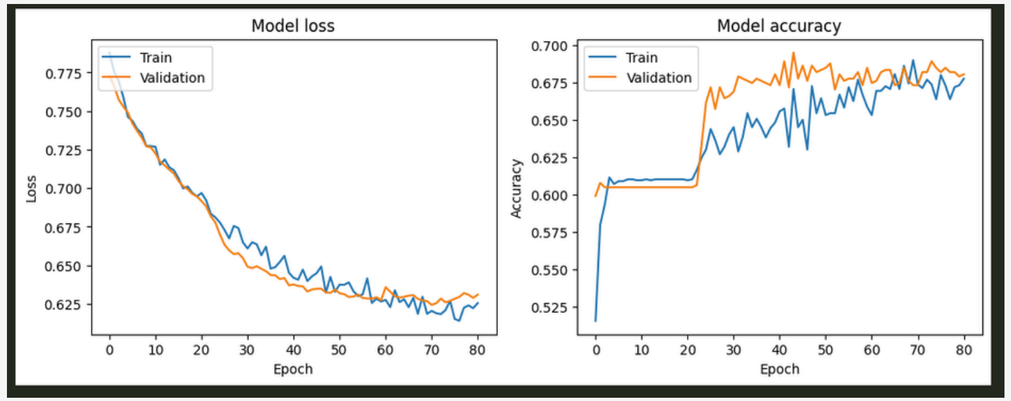
Choice of the Model.



4th Model was taken

AFTER CLEARLY LOOKING AT THE DIFFERENT MODELS, WE DECIDED TO MOVE ON WITH THE 4TH MODEL FOR OUR WORK.

Choice of the Model.



4th Model was taken

AFTER CLEARLY LOOKING AT THE DIFFERENT MODELS, WE DECIDED TO MOVE ON WITH THE 4TH MODEL FOR OUR WORK.