INTRODUCTION

Fire is a natural disaster that can be caused by many different reasons. An innovative sound wave fire-extinguishing system was created and firefighting tests were performed. Machine learning algorithms were applied to data crated by performing several tests for classification of extinction and non-extinction states of the flame.

The acoustic extinguisher works by using sound waves—a type of pressure wave—to push oxygen away from the source of a flame and spread it over a larger surface area. These actions break the fire combustion triangle made up of heat, fuel, and oxygen, the three elements required for a fire to burn. The acoustic fire extinguisher puts out flames using low frequency bass (30 to 60Hz) without relying on water or chemicals.[1]

By hitting fire with the low-frequency sound waves in the 30 to 60 hertz range, the device separates oxygen from fuel. The pressure wave is going back and forth, and that agitates where the air is. That specific space is enough to keep the fire from reigniting.[2]

Dataset information:- Acoustic_Extinguisher_Fire_Dataset.csv

The dataset consists of 6 predictors and a response variable, STATUS. It is formulated as a binary classification problem.

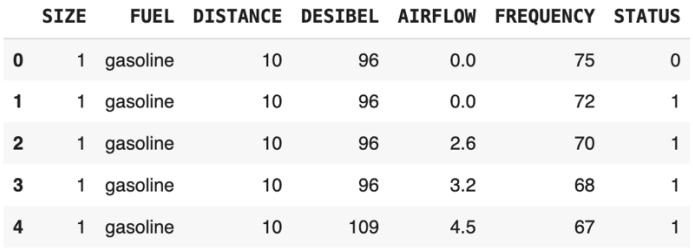
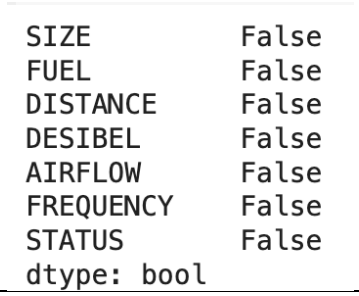
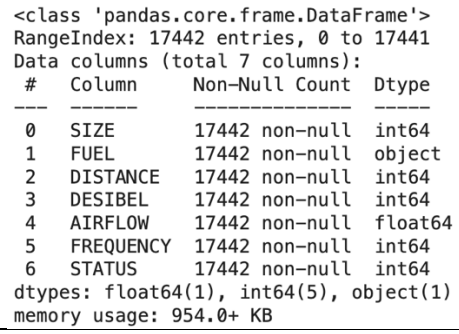
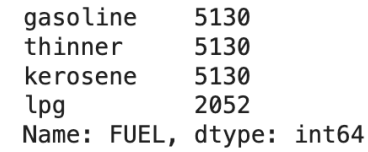
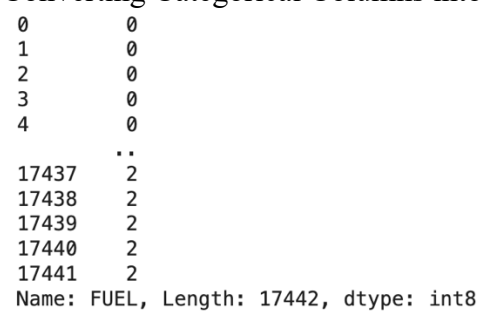
The dataset was obtained as a result of the extinguishing tests of four different fuel flames with a sound wave extinguishing system.

Throughout the flame extinguishing experiments, the data obtained from each measurement device was recorded and a dataset was created. The dataset includes the features of fuel container size representing the flame size, fuel type, frequency, decibel, distance, airflow and flame extinction. Accordingly, 6 input features and 1 output feature will be used in models.

The status property (flame extinction or non-extinction states) can be predicted by using six features in the dataset. Status and fuel features are categorical, while other features are numerical. 8,759 of the 17,442 test results are the non-extinguishing state of the flame. 8,683 of them are the extinction state of the flame. According to these numbers, it can be said that the class distribution of the dataset is almost equal.[3]

OBSERVATION TABLE

Observation number	Output	Remark\ Interpretation
	Part 1: Baseline Model	
1	Task Description: The given task is of binary classification. We have been data from fires caused by different sources i.e. fuels and whether synthesized sound waves generated from a	If the fire is extinguished, that data instance is added to class 1, else added to class 0. Hence the given task is of Binary

	computer would extinguish the fire.	Classification.
2	Type of data and description: The variable, df is of type: <class 'pandas.core.frame.DataFrame'> (17442, 7) 	The data given is extracted from a CSV file in the form of a Pandas Data frame. There are 17442 data instances with 7 columns.
3	Column names and predictor values: `Index(['SIZE', 'FUEL', 'DISTANCE', 'DESIBEL', 'AIRFLOW', 'FREQUENCY', 'STATUS'],`	
4	Missing Values: 	No missing or NAN values which shows that the data provided is cleaned and little to no pre-processing is required.
5	Type of Data predictors: 	The FUEL type and STATUS are categorical and the rest of the predictors and integer or floating-point values.
6	Balancing the dataset: 	The given dataset is balanced as it has equal entries of all types of fuels and thus the results will not be biased towards one specific type of fuel.
7	Converting Categorical Columns into Numeric Form 	We have used One-Hot-Encoding to convert the categorical columns into Numeric form to further process the data. Each Fuel type has been replaced by its equivalent number.

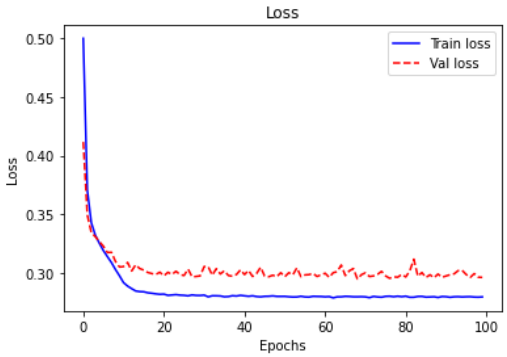
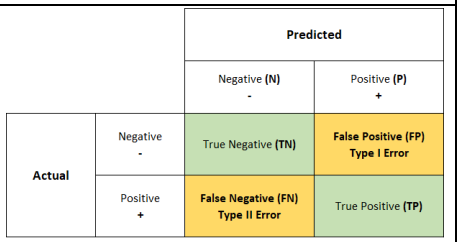
	SIZE	FUEL	DISTANCE	DESIBEL	AIRFLOW	FREQUENCY	STATUS
0	1	gasoline	10	96	0.0	75	0
1	1	gasoline	10	96	0.0	72	1
2	1	gasoline	10	96	2.6	70	1
3	1	gasoline	10	96	3.2	68	1
4	1	gasoline	10	109	4.5	67	1

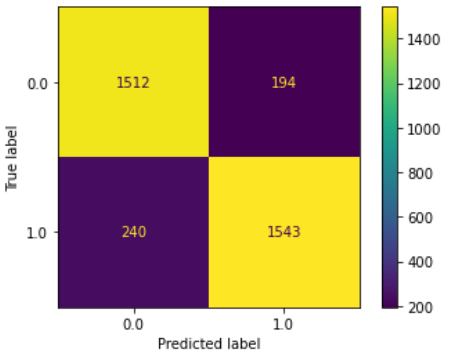
```
SIZE          False
FUEL          False
DISTANCE      False
DESIBEL       False
AIRFLOW       False
FREQUENCY     False
STATUS        False
dtype: bool
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17442 entries, 0 to 17441
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   SIZE       17442 non-null  int64
 1   FUEL       17442 non-null  object
 2   DISTANCE   17442 non-null  int64
 3   DESIBEL    17442 non-null  int64
 4   AIRFLOW    17442 non-null  float64
 5   FREQUENCY  17442 non-null  int64
 6   STATUS     17442 non-null  int64
dtypes: float64(1), int64(5), object(1)
memory usage: 954.0+ KB
```

```
gasoline    5130
thinner     5130
kerosene    5130
lpg         2052
Name: FUEL, dtype: int64
```

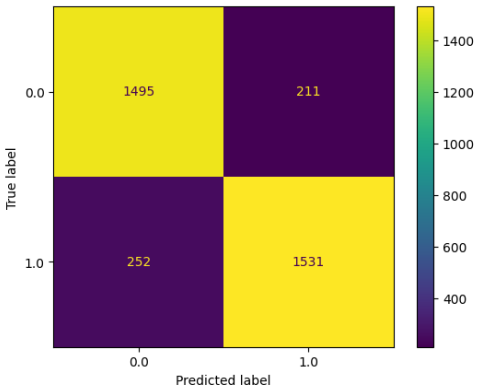
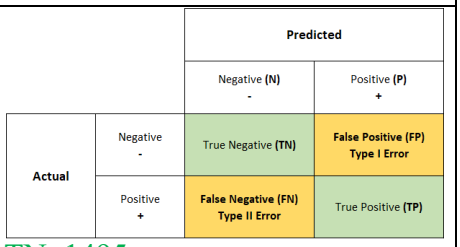
```
0        0
1        0
2        0
3        0
4        0
        ..
17437    2
17438    2
17439    2
17440    2
17441    2
Name: FUEL, Length: 17442, dtype: int8
```

8	The predictor input and response output have been split into an 80-20 Test and Train subset.	
9	Choosing the model and network architecture: We have implemented a hidden layer of 4 neurons containing the RELU activation function and an output layer of 1 neuron with the SIGMOID activation function	As the given task is of binary classification, we have deployed only one neuron in the output layer using the sigmoidal activation function. Both layers are dense and fully connected.
10	Number of trainable parameters: Model: "sequential" Layer (type) | Output Shape | Param # dense (Dense) | (None, 4) | 28 dense_1 (Dense) | (None, 1) | 5 Total params: 33 (132.00 Byte) Trainable params: 33 (132.00 Byte) Non-trainable params: 0 (0.00 Byte)	There are a total of 33 trainable parameters.
11	Initiating the Model: Optimizer: Adam Loss: Binary Cross Entropy Batch size: 8 Epochs: 100	In the current state, The Adam optimizer provides the best efficiency in terms of locating the global minima. We have used a batch size of 8 and have used 100 iterations.
12	Loss vs Epochs: 	The plot shows how the loss has reduced over multiple epochs and has settled at 0.30 for the training set and 0.32 for the validation set.
13	Train Accuracy : 0.90 Test Accuracy : 0.90	We have obtained an accuracy of 90 percent on both the test and validation set.
14	Confusion Matrix:	 The confusion matrix is a plot as depicted above. Our values are as follows:

		TN: 1512 FP: 194 FN:240 TP:1543
	PART 2: **Hyperparameter Tuning using KerasClassifier from scikeras and GridSearch from Sklearn**	

15	Batch Size Optimization:
	`#Observation21:Note hyperparameters setting that gave the best`
	`#result along with the best accuracy score`
	
	`print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))`
	`means = grid_result.cv_results_['mean_test_score']`
	`stds = grid_result.cv_results_['std_test_score']`
	`params = grid_result.cv_results_['params']`
	
	`#Compute predictions on Test data`
	
	`Y_preds = grid.predict(X_test)`
	
	`Best: 0.872429 using {'batch_size': 10, 'epochs': 5}`
	`349/349 [==============================] - 1s 1ms/step`
		By using Grid Search on the test data, we have found that the best batch size is 10 with epochs 5

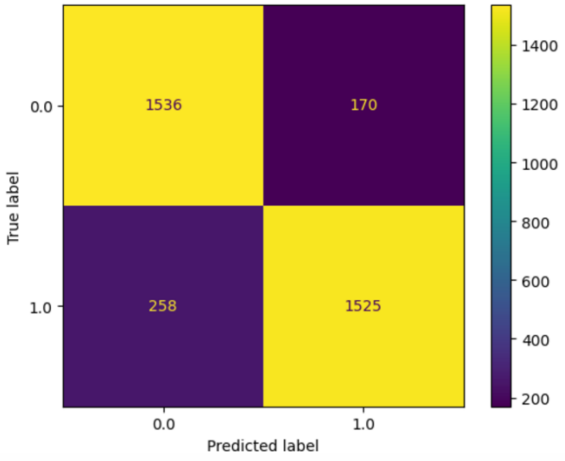
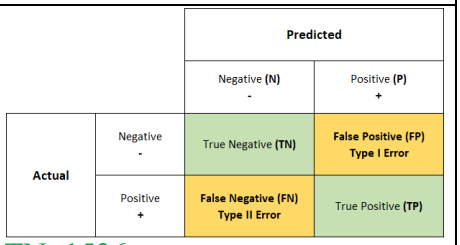
16	Model Test and Train Accuracy:
	`349/349 [==============================] - 1s 1ms/step`
	`Test Accuracy : 0.87`
	`1396/1396 [==============================] - 2s 1ms/step`
	`Train Accuracy : 0.87`
		Applying the Ideal Batch Parameters and epochs to the test and training dataset.

17	Confusion Matrix with Grid Search: 	 TN: 1495 FP: 211 FN:252 TP:1531
	PART 3: Using Grid Search to find ideal Optimizer.	
18	In this sub part, we have used GridSearch from Sklearn to obtain the model optimizer.	

19	Ideal Optimizer:
	`Best: 0.867389 using {'optimizer': 'Adam'}`
	`0.740798 (0.168747) with: {'optimizer': 'SGD'}`
	`0.858044 (0.002517) with: {'optimizer': 'RMSprop'}`
	`0.867389 (0.003673) with: {'optimizer': 'Adam'}`
		From a list of 3 optimizers, SGD, RMSProp and Adam, after using Grid Search to obtain model parameters, we have inferred that Adam is the best optimizer for

		our dataset.

20	Test and training accuracy with Adam:
	`Test Accuracy : 0.88`
	`Train Accuracy : 0.88`	

21	Confusion Matrix: 	 TN: 1536 FP: 170 FN:258 TP:1525

CONCLUSION

In this assignment, we have been provided with a dataset which contains the details of acoustic fire extinguishing. The given task if of Binary Classification. The fire has been lighted using different fuels and we have classified binary 1 if the sound waves produced have extinguished the fire or binary 0 if the fire has not been extinguished.

In Part 1, we have performed the baseline modelling. As the given data was already preprocessed, we have directly started modelling the dataset.

We have deployed an 80-20 data split and used one hidden layer with 4 neurons and RELU activation function. In the output layer we have used the sigmoidal activation function.

We obtained an accuracy of 90% in both the training and test sets. The confusion matrix states that we have obtained TN: 1512, FP: 194, FN:240, TP:1543

In part 2, we have used Grid Search from Sci-Kit learn to find the ideal model parameters. We have found the model batch parameters along with the model epochs.

We have obtained an accuracy of 87% accuracy in both the training and test sets. The confusion matrix states that we have obtained TN: 1495, FP: 211, FN:252, TP:1531

In part 3, we have used Grid Search from Sci-Kit learn to find the ideal model parameters. We have found the model optimizer function.

We have obtained an accuracy of 88% accuracy in both the training and test sets. The confusion matrix states that we have obtained TN: 1536, FP: 170, FN:258, TP:1525.

CITATIONS

[1] https://www.dell.com/en-us/perspectives/fighting-fire-with-bass-using-sound-waves-to-drown-flames/

[2] http://ir.aiktclibrary.org:8080/xmlui/bitstream/handle/123456789/3057/GRP25-%20Fire%20Extinguisher.pdf?sequence=1&isAllowed=y#:~:text=By%20hitting%20fire%20with%20the,keep%20the%20fire%20from%20reigniting.

[3] Research Paper provided.