

```
# Installing Theano
# pip install --upgrade --no-deps
git+git://github.com/Theano/Theano.git

# Installing Tensorflow
# pip install tensorflow

# Installing Keras
# pip install --upgrade keras

# importing the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Loading the data
dataset = pd.read_csv('sonar.csv', header = None)
X = dataset.iloc[:, 0:60].values
y = dataset.iloc[:, 60].values

# Feature Encoding
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
encoder = LabelEncoder()
y = encoder.fit_transform(y)

# Splitting into training and test sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.25, random_state = 0)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# Building an ANN
# Importing Keras Libraries and packages
import keras
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout

# Initialize the ANN
classifier = Sequential()

# Build the input and hidden layers with dropout
classifier.add(Dense(units = 32, activation = 'relu',
kernel_initializer = 'uniform', input_dim = 60))
classifier.add(Dropout(rate = 0.1))

# Adding the second hidden layer
classifier.add(Dense(units = 32, activation = 'relu',
kernel_initializer = 'uniform'))
classifier.add(Dropout(rate = 0.1))
```

```
# Adding an output layer
classifier.add(Dense(units = 1, kernel_initializer = 'uniform',
activation = 'sigmoid'))

# Compiling the ANN
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy',
metrics = ['accuracy'])

# Fitting ANN to training set
classifier.fit(X_train, y_train, batch_size = 10, epochs = 10)

# Predicting on test set
y_pred = classifier.predict(X_test) > 0.5

# Evaluating using confusion matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```