

```
In [1]: import csv
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('seaborn')
```

```
In [2]: car_df = pd.read_csv('./data.csv')
```

```
In [3]: car_df.shape
```

```
Out[3]: (11914, 16)
```

```
In [4]: car_df.dtypes
```

```
Out[4]: Make                object
Model                object
Year                 int64
EngineFuelType       object
EngineHP             float64
EngineCylinders      float64
TransmissionType     object
Driven_Wheels        object
NumberofDoors        float64
MarketCategory       object
VehicleSize          object
VehicleStyle         object
highwayMPG           int64
citympg              int64
Popularity            int64
MSRP                 int64
dtype: object
```

```
In [6]: car_df = car_df.dropna()
car_df = pd.get_dummies(car_df, columns=['Make'])
car_df = pd.get_dummies(car_df, columns=['EngineFuelType'])
car_df = pd.get_dummies(car_df, columns=['TransmissionType'])
car_df = pd.get_dummies(car_df, columns=['Driven_Wheels'])
car_df = pd.get_dummies(car_df, columns=['VehicleSize'])
car_df = pd.get_dummies(car_df, columns=['VehicleStyle'])
```

```
In [7]: car_df["Price"] = ""
car_df.loc[car_df['MSRP'] <= 50000 , "Price"] = 0
car_df.loc[car_df['MSRP'] > 50000, "Price"] = 1

car_df['Price'] = pd.to_numeric(car_df['Price'], errors='coerce')
car_df = car_df.dropna(subset=['Price'])
car_df['Price'] = car_df['Price'].astype(int)
```

```
In [8]: car_df['EngineHP'] = pd.to_numeric(car_df['EngineHP'], errors='coerce')
car_df = car_df.dropna(subset=['EngineHP'])
car_df['EngineHP'] = car_df['EngineHP'].astype(int)

car_df['EngineCylinders'] = pd.to_numeric(car_df['EngineCylinders'], errors='coerce')
car_df = car_df.dropna(subset=['EngineCylinders'])
car_df['EngineCylinders'] = car_df['EngineCylinders'].astype(int)

car_df['NumberofDoors'] = pd.to_numeric(car_df['NumberofDoors'], errors='coerce')
car_df = car_df.dropna(subset=['NumberofDoors'])
car_df['NumberofDoors'] = car_df['NumberofDoors'].astype(int)
```

```
In [9]: x = car_df.drop(['Model', 'MSRP', 'Price', 'Popularity', 'MarketCategory'],axis=1)
y = car_df['Price'].values.reshape(-1,1)
```

```
In [10]: x.head()
```

```
Out[10]:
```

	Year	EngineHP	EngineCylinders	NumberofDoors	highwayMPG	citympg	Make_Acura	Make_Alfa Romeo	Make_Aston Martin	Make_Audi	...	VehicleSt
0	2011	335	6	2	26	19	0	0	0	0	...	
1	2011	300	6	2	28	19	0	0	0	0	...	
2	2011	300	6	2	28	20	0	0	0	0	...	
3	2011	230	6	2	28	18	0	0	0	0	...	
4	2011	230	6	2	28	18	0	0	0	0	...	

5 rows × 89 columns

```
In [11]: y.shape
```

Out[11]: (8084, 1)

In [13]: **from** sklearn.preprocessing **import** MinMaxScaler

```
MinMax_scaler = MinMaxScaler()
X_scaled = MinMax_scaler.fit_transform(x)
y_scaled = MinMax_scaler.fit_transform(y)
```

In [14]: **from** sklearn.model\_selection **import** train\_test\_split  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y\_scaled, test\_size = 0.30)

```
print("x_train: ",X_train.shape)
print("y_train: ",y_train.shape)
print("x_test: ",X_test.shape)
print("y_test: ",y_test.shape)
```

```
x_train: (5658, 89)
y_train: (5658, 1)
x_test: (2426, 89)
y_test: (2426, 1)
```

In [15]: **import** tensorflow.keras  
**import** keras  
**from** keras.models **import** Sequential  
**from** keras.layers **import** Dense, Dropout

```
model = Sequential()
model.add(Dense(45,input_dim =89, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(45, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(25, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 45)	4050
dropout (Dropout)	(None, 45)	0
dense_1 (Dense)	(None, 45)	2070
dropout_1 (Dropout)	(None, 45)	0
dense_2 (Dense)	(None, 25)	1150
dense_3 (Dense)	(None, 1)	26
Total params: 7,296		
Trainable params: 7,296		
Non-trainable params: 0		

In [16]: *# Compile Model*  
adm = keras.optimizers.Adam(lr=0.001)  
sgd = keras.optimizers.SGD(lr=0.001, decay=1e-6, momentum=0.5, nesterov=True)

```
model.compile(optimizer=adm, loss='mean_absolute_error',metrics='accuracy')
```

```
epochs_hist = model.fit(X_train, y_train, epochs=100, batch_size=128, verbose=1, validation_split=0.10)
```

```
Epoch 1/100
40/40 [=====] - 2s 39ms/step - loss: 0.4096 - accuracy: 0.7147 - val_loss: 0.2155 - val_accuracy: 0.7933
Epoch 2/100
40/40 [=====] - 0s 4ms/step - loss: 0.2435 - accuracy: 0.7633 - val_loss: 0.2052 - val_accuracy: 0.7933
Epoch 3/100
40/40 [=====] - 0s 4ms/step - loss: 0.2373 - accuracy: 0.7610 - val_loss: 0.1989 - val_accuracy: 0.7933
Epoch 4/100
40/40 [=====] - 0s 4ms/step - loss: 0.2249 - accuracy: 0.7613 - val_loss: 0.1811 - val_accuracy: 0.7933
Epoch 5/100
40/40 [=====] - 0s 3ms/step - loss: 0.1979 - accuracy: 0.7971 - val_loss: 0.1526 - val_accuracy: 0.8922
Epoch 6/100
40/40 [=====] - 0s 4ms/step - loss: 0.1673 - accuracy: 0.8488 - val_loss: 0.1153 - val_accuracy: 0.9028
Epoch 7/100
40/40 [=====] - 0s 4ms/step - loss: 0.1447 - accuracy: 0.8695 - val_loss: 0.0956 - val_accuracy: 0.9117
Epoch 8/100
40/40 [=====] - 0s 3ms/step - loss: 0.1213 - accuracy: 0.8877 - val_loss: 0.0730 - val_accuracy: 0.9311
```

Epoch 9/100  
40/40 [=====] - 0s 5ms/step - loss: 0.1033 - accuracy: 0.9058 - val\_loss: 0.0698 - val  
\_accuracy: 0.9346  
Epoch 10/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0981 - accuracy: 0.9047 - val\_loss: 0.0623 - val  
\_accuracy: 0.9382  
Epoch 11/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0927 - accuracy: 0.9130 - val\_loss: 0.0626 - val  
\_accuracy: 0.9417  
Epoch 12/100  
40/40 [=====] - 0s 6ms/step - loss: 0.0853 - accuracy: 0.9164 - val\_loss: 0.0565 - val  
\_accuracy: 0.9470  
Epoch 13/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0894 - accuracy: 0.9137 - val\_loss: 0.0584 - val  
\_accuracy: 0.9452  
Epoch 14/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0807 - accuracy: 0.9212 - val\_loss: 0.0545 - val  
\_accuracy: 0.9470  
Epoch 15/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0854 - accuracy: 0.9178 - val\_loss: 0.0542 - val  
\_accuracy: 0.9452  
Epoch 16/100  
40/40 [=====] - 0s 3ms/step - loss: 0.0810 - accuracy: 0.9213 - val\_loss: 0.0530 - val  
\_accuracy: 0.9505  
Epoch 17/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0785 - accuracy: 0.9245 - val\_loss: 0.0547 - val  
\_accuracy: 0.9452  
Epoch 18/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0751 - accuracy: 0.9282 - val\_loss: 0.0538 - val  
\_accuracy: 0.9452  
Epoch 19/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0701 - accuracy: 0.9315 - val\_loss: 0.0531 - val  
\_accuracy: 0.9488  
Epoch 20/100  
40/40 [=====] - 0s 3ms/step - loss: 0.0723 - accuracy: 0.9291 - val\_loss: 0.0545 - val  
\_accuracy: 0.9470  
Epoch 21/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0700 - accuracy: 0.9309 - val\_loss: 0.0535 - val  
\_accuracy: 0.9452  
Epoch 22/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0668 - accuracy: 0.9339 - val\_loss: 0.0494 - val  
\_accuracy: 0.9523  
Epoch 23/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0582 - accuracy: 0.9420 - val\_loss: 0.0502 - val  
\_accuracy: 0.9523  
Epoch 24/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0669 - accuracy: 0.9354 - val\_loss: 0.0501 - val  
\_accuracy: 0.9505  
Epoch 25/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0642 - accuracy: 0.9373 - val\_loss: 0.0479 - val  
\_accuracy: 0.9541  
Epoch 26/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0623 - accuracy: 0.9378 - val\_loss: 0.0501 - val  
\_accuracy: 0.9523  
Epoch 27/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0628 - accuracy: 0.9379 - val\_loss: 0.0511 - val  
\_accuracy: 0.9505  
Epoch 28/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0650 - accuracy: 0.9354 - val\_loss: 0.0512 - val  
\_accuracy: 0.9505  
Epoch 29/100  
40/40 [=====] - 0s 3ms/step - loss: 0.0603 - accuracy: 0.9405 - val\_loss: 0.0497 - val  
\_accuracy: 0.9470  
Epoch 30/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0597 - accuracy: 0.9418 - val\_loss: 0.0522 - val  
\_accuracy: 0.9488  
Epoch 31/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0648 - accuracy: 0.9376 - val\_loss: 0.0535 - val  
\_accuracy: 0.9435  
Epoch 32/100  
40/40 [=====] - 0s 3ms/step - loss: 0.0649 - accuracy: 0.9357 - val\_loss: 0.0481 - val  
\_accuracy: 0.9505  
Epoch 33/100  
40/40 [=====] - 0s 3ms/step - loss: 0.0646 - accuracy: 0.9358 - val\_loss: 0.0488 - val  
\_accuracy: 0.9523  
Epoch 34/100  
40/40 [=====] - 0s 3ms/step - loss: 0.0596 - accuracy: 0.9417 - val\_loss: 0.0487 - val  
\_accuracy: 0.9505  
Epoch 35/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0607 - accuracy: 0.9397 - val\_loss: 0.0506 - val  
\_accuracy: 0.9488  
Epoch 36/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0586 - accuracy: 0.9430 - val\_loss: 0.0496 - val  
\_accuracy: 0.9523  
Epoch 37/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0556 - accuracy: 0.9456 - val\_loss: 0.0484 - val  
\_accuracy: 0.9505  
Epoch 38/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0571 - accuracy: 0.9443 - val\_loss: 0.0475 - val

```
_accuracy: 0.9541
Epoch 39/100
40/40 [=====] - 0s 4ms/step - loss: 0.0591 - accuracy: 0.9417 - val_loss: 0.0487 - val
_accuracy: 0.9505
Epoch 40/100
40/40 [=====] - 0s 5ms/step - loss: 0.0544 - accuracy: 0.9456 - val_loss: 0.0466 - val
_accuracy: 0.9541
Epoch 41/100
40/40 [=====] - 0s 5ms/step - loss: 0.0567 - accuracy: 0.9449 - val_loss: 0.0478 - val
_accuracy: 0.9523
Epoch 42/100
40/40 [=====] - 0s 5ms/step - loss: 0.0597 - accuracy: 0.9420 - val_loss: 0.0456 - val
_accuracy: 0.9558
Epoch 43/100
40/40 [=====] - 0s 4ms/step - loss: 0.0556 - accuracy: 0.9450 - val_loss: 0.0494 - val
_accuracy: 0.9505
Epoch 44/100
40/40 [=====] - 0s 6ms/step - loss: 0.0613 - accuracy: 0.9384 - val_loss: 0.0473 - val
_accuracy: 0.9541
Epoch 45/100
40/40 [=====] - 0s 4ms/step - loss: 0.0597 - accuracy: 0.9406 - val_loss: 0.0447 - val
_accuracy: 0.9558
Epoch 46/100
40/40 [=====] - 0s 3ms/step - loss: 0.0584 - accuracy: 0.9424 - val_loss: 0.0452 - val
_accuracy: 0.9558
Epoch 47/100
40/40 [=====] - 0s 4ms/step - loss: 0.0580 - accuracy: 0.9424 - val_loss: 0.0449 - val
_accuracy: 0.9576
Epoch 48/100
40/40 [=====] - 0s 5ms/step - loss: 0.0556 - accuracy: 0.9468 - val_loss: 0.0443 - val
_accuracy: 0.9576
Epoch 49/100
40/40 [=====] - 0s 5ms/step - loss: 0.0541 - accuracy: 0.9478 - val_loss: 0.0457 - val
_accuracy: 0.9541
Epoch 50/100
40/40 [=====] - 0s 4ms/step - loss: 0.0585 - accuracy: 0.9429 - val_loss: 0.0448 - val
_accuracy: 0.9576
Epoch 51/100
40/40 [=====] - 0s 4ms/step - loss: 0.0584 - accuracy: 0.9415 - val_loss: 0.0471 - val
_accuracy: 0.9523
Epoch 52/100
40/40 [=====] - 0s 4ms/step - loss: 0.0544 - accuracy: 0.9469 - val_loss: 0.0501 - val
_accuracy: 0.9505
Epoch 53/100
40/40 [=====] - 0s 4ms/step - loss: 0.0558 - accuracy: 0.9452 - val_loss: 0.0461 - val
_accuracy: 0.9541
Epoch 54/100
40/40 [=====] - 0s 5ms/step - loss: 0.0556 - accuracy: 0.9447 - val_loss: 0.0443 - val
_accuracy: 0.9558
Epoch 55/100
40/40 [=====] - 0s 4ms/step - loss: 0.0581 - accuracy: 0.9434 - val_loss: 0.0461 - val
_accuracy: 0.9523
Epoch 56/100
40/40 [=====] - 0s 5ms/step - loss: 0.0538 - accuracy: 0.9466 - val_loss: 0.0455 - val
_accuracy: 0.9541
Epoch 57/100
40/40 [=====] - 0s 4ms/step - loss: 0.0605 - accuracy: 0.9401 - val_loss: 0.0417 - val
_accuracy: 0.9629
Epoch 58/100
40/40 [=====] - 0s 7ms/step - loss: 0.0514 - accuracy: 0.9494 - val_loss: 0.0449 - val
_accuracy: 0.9576
Epoch 59/100
40/40 [=====] - 0s 5ms/step - loss: 0.0572 - accuracy: 0.9429 - val_loss: 0.0438 - val
_accuracy: 0.9558
Epoch 60/100
40/40 [=====] - 0s 4ms/step - loss: 0.0552 - accuracy: 0.9453 - val_loss: 0.0447 - val
_accuracy: 0.9558
Epoch 61/100
40/40 [=====] - 0s 5ms/step - loss: 0.0474 - accuracy: 0.9533 - val_loss: 0.0416 - val
_accuracy: 0.9611
Epoch 62/100
40/40 [=====] - 0s 4ms/step - loss: 0.0502 - accuracy: 0.9495 - val_loss: 0.0443 - val
_accuracy: 0.9558
Epoch 63/100
40/40 [=====] - 0s 7ms/step - loss: 0.0493 - accuracy: 0.9505 - val_loss: 0.0444 - val
_accuracy: 0.9558
Epoch 64/100
40/40 [=====] - 0s 5ms/step - loss: 0.0480 - accuracy: 0.9513 - val_loss: 0.0412 - val
_accuracy: 0.9594
Epoch 65/100
40/40 [=====] - 0s 4ms/step - loss: 0.0483 - accuracy: 0.9522 - val_loss: 0.0428 - val
_accuracy: 0.9558
Epoch 66/100
40/40 [=====] - 0s 5ms/step - loss: 0.0545 - accuracy: 0.9464 - val_loss: 0.0410 - val
_accuracy: 0.9594
Epoch 67/100
40/40 [=====] - 0s 4ms/step - loss: 0.0523 - accuracy: 0.9480 - val_loss: 0.0430 - val
_accuracy: 0.9576
Epoch 68/100
```

40/40 [=====] - 0s 4ms/step - loss: 0.0487 - accuracy: 0.9514 - val\_loss: 0.0441 - val  
\_accuracy: 0.9558  
Epoch 69/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0503 - accuracy: 0.9494 - val\_loss: 0.0445 - val  
\_accuracy: 0.9558  
Epoch 70/100  
40/40 [=====] - 0s 7ms/step - loss: 0.0450 - accuracy: 0.9561 - val\_loss: 0.0431 - val  
\_accuracy: 0.9558  
Epoch 71/100  
40/40 [=====] - 0s 6ms/step - loss: 0.0470 - accuracy: 0.9541 - val\_loss: 0.0430 - val  
\_accuracy: 0.9558  
Epoch 72/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0403 - accuracy: 0.9607 - val\_loss: 0.0458 - val  
\_accuracy: 0.9541  
Epoch 73/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0450 - accuracy: 0.9545 - val\_loss: 0.0441 - val  
\_accuracy: 0.9558  
Epoch 74/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0450 - accuracy: 0.9557 - val\_loss: 0.0430 - val  
\_accuracy: 0.9558  
Epoch 75/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0409 - accuracy: 0.9603 - val\_loss: 0.0436 - val  
\_accuracy: 0.9558  
Epoch 76/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0437 - accuracy: 0.9572 - val\_loss: 0.0446 - val  
\_accuracy: 0.9558  
Epoch 77/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0429 - accuracy: 0.9575 - val\_loss: 0.0466 - val  
\_accuracy: 0.9541  
Epoch 78/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0469 - accuracy: 0.9532 - val\_loss: 0.0475 - val  
\_accuracy: 0.9523  
Epoch 79/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0461 - accuracy: 0.9543 - val\_loss: 0.0417 - val  
\_accuracy: 0.9594  
Epoch 80/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0480 - accuracy: 0.9510 - val\_loss: 0.0492 - val  
\_accuracy: 0.9505  
Epoch 81/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0431 - accuracy: 0.9577 - val\_loss: 0.0440 - val  
\_accuracy: 0.9558  
Epoch 82/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0452 - accuracy: 0.9560 - val\_loss: 0.0438 - val  
\_accuracy: 0.9576  
Epoch 83/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0453 - accuracy: 0.9562 - val\_loss: 0.0488 - val  
\_accuracy: 0.9505  
Epoch 84/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0454 - accuracy: 0.9557 - val\_loss: 0.0446 - val  
\_accuracy: 0.9558  
Epoch 85/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0420 - accuracy: 0.9580 - val\_loss: 0.0414 - val  
\_accuracy: 0.9611  
Epoch 86/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0500 - accuracy: 0.9500 - val\_loss: 0.0433 - val  
\_accuracy: 0.9558  
Epoch 87/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0445 - accuracy: 0.9562 - val\_loss: 0.0435 - val  
\_accuracy: 0.9576  
Epoch 88/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0415 - accuracy: 0.9589 - val\_loss: 0.0393 - val  
\_accuracy: 0.9611  
Epoch 89/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0404 - accuracy: 0.9608 - val\_loss: 0.0442 - val  
\_accuracy: 0.9576  
Epoch 90/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0505 - accuracy: 0.9497 - val\_loss: 0.0427 - val  
\_accuracy: 0.9558  
Epoch 91/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0424 - accuracy: 0.9583 - val\_loss: 0.0415 - val  
\_accuracy: 0.9576  
Epoch 92/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0420 - accuracy: 0.9587 - val\_loss: 0.0441 - val  
\_accuracy: 0.9558  
Epoch 93/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0389 - accuracy: 0.9612 - val\_loss: 0.0422 - val  
\_accuracy: 0.9576  
Epoch 94/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0419 - accuracy: 0.9593 - val\_loss: 0.0421 - val  
\_accuracy: 0.9576  
Epoch 95/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0437 - accuracy: 0.9561 - val\_loss: 0.0410 - val  
\_accuracy: 0.9594  
Epoch 96/100  
40/40 [=====] - 0s 4ms/step - loss: 0.0453 - accuracy: 0.9544 - val\_loss: 0.0371 - val  
\_accuracy: 0.9629  
Epoch 97/100  
40/40 [=====] - 0s 5ms/step - loss: 0.0426 - accuracy: 0.9575 - val\_loss: 0.0419 - val  
\_accuracy: 0.9576

```
Epoch 98/100
40/40 [=====] - 0s 4ms/step - loss: 0.0421 - accuracy: 0.9579 - val_loss: 0.0426 - val_accuracy: 0.9558
Epoch 99/100
40/40 [=====] - 0s 4ms/step - loss: 0.0438 - accuracy: 0.9566 - val_loss: 0.0366 - val_accuracy: 0.9647
Epoch 100/100
40/40 [=====] - 0s 4ms/step - loss: 0.0436 - accuracy: 0.9571 - val_loss: 0.0409 - val_accuracy: 0.9594
```

```
In [18]: from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

y_pred = model.predict_classes(X_test, verbose = 1)
correct = np.sum(y_pred == y_test)

print ('Test Accuracy: ', correct/float(y_test.shape[0])*100.0, '%')

print('Macro Precision: {:.2f}'.format(precision_score(y_test, y_pred, average='macro')*100))
print('Macro Recall: {:.2f}'.format(recall_score(y_test, y_pred, average='macro')*100))
print('Macro F1-score: {:.2f}\n'.format(f1_score(y_test, y_pred, average='macro')*100))
```

```
76/76 [=====] - 0s 1ms/step
Test Accuracy: 95.42456718878813 %
Macro Precision: 95.06
Macro Recall: 92.07
Macro F1-score: 93.45
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js