In [1]:
```python
#Importing the Libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```python
df=pd.read_csv('data.csv')
```

In [3]:
```python
df.head(5)
```

Out[3]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothne
0	842302	M	17.99	10.38	122.80	1001.0	
1	842517	M	20.57	17.77	132.90	1326.0	
2	84300903	M	19.69	21.25	130.00	1203.0	
3	84348301	M	11.42	20.38	77.58	386.1	
4	84358402	M	20.29	14.34	135.10	1297.0	

5 rows × 33 columns

In [4]:
```python
df.drop('id',axis=1,inplace=True)
df.drop('Unnamed: 32',axis=1,inplace=True)
```

In [6]:
```python
df.shape
```

Out[6]:
```
(569, 31)
```

In [7]:
```python
df.diagnosis.unique()
```

Out[7]:
```
array(['M', 'B'], dtype=object)
```

In [8]:
```python
df['diagnosis'] = df['diagnosis'].map({'M':1,'B':0})
df.head()
```

Out[8]:

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean
0	1	17.99	10.38	122.80	1001.0	0.11840
1	1	20.57	17.77	132.90	1326.0	0.08474
2	1	19.69	21.25	130.00	1203.0	0.10960
3	1	11.42	20.38	77.58	386.1	0.14250
4	1	20.29	14.34	135.10	1297.0	0.10030

5 rows × 31 columns

In [10]:
```python
#Checking out statistics (mean, median standard deviation)
df.describe()
```

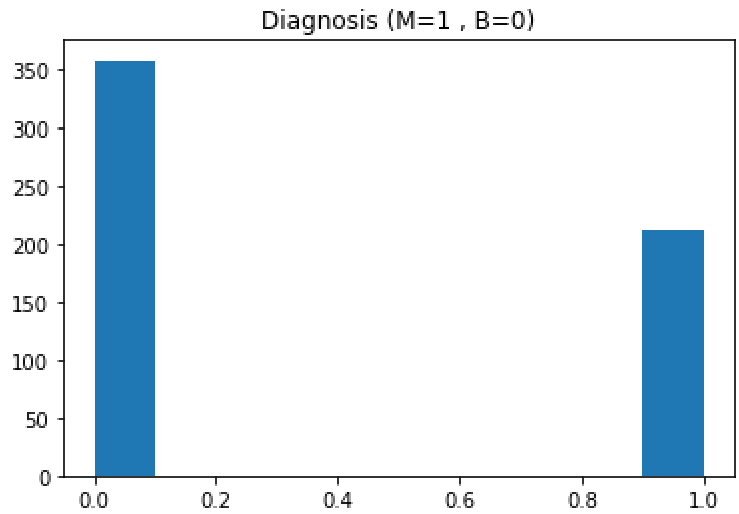
Out[10]:

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_r

count	569.000000	569.000000	569.000000	569.000000	569.000000	569.00
mean	0.372583	14.127292	19.289649	91.969033	654.889104	0.09
std	0.483918	3.524049	4.301036	24.298981	351.914129	0.01
min	0.000000	6.981000	9.710000	43.790000	143.500000	0.05
25%	0.000000	11.700000	16.170000	75.170000	420.300000	0.08
50%	0.000000	13.370000	18.840000	86.240000	551.100000	0.09
75%	1.000000	15.780000	21.800000	104.100000	782.700000	0.10
max	1.000000	28.110000	39.280000	188.500000	2501.000000	0.16

8 rows × 31 columns

In [11]:
```
df.describe()
plt.hist(df['diagnosis'])
plt.title('Diagnosis (M=1 , B=0)')
plt.show()
```



In [12]:
```
features_mean=list(df.columns[1:11])

# split dataframe into two based on diagnosis
dfM=df[df['diagnosis'] ==1]
dfB=df[df['diagnosis'] ==0]
```

In [15]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler

def getNormalizedData(X):

    # fit scaler on training data
    norm = MinMaxScaler().fit(X)

    # transform training data
    X_train_norm = norm.transform(X)
    X_train_norm = pd.DataFrame(X_train_norm, columns=X.columns.values)

    return X_train_norm
```

In [17]:
```
#'Diagnosis' is the target variable, change as applicable
y = df.diagnosis
X = df.drop('diagnosis',1)
```

```
X = getNormalizedData(X)
```

In [18]:
```
train_X, val_X, train_y, val_y = train_test_split(X, y, random_state=42,
```

In [19]:
```
def get_nn_simplemodel(n_inputs=22, n_outputs=1, optimizerinput='adam'):
    # create model
    model = Sequential()

    model.add(layers.Dense(n_inputs, input_dim=n_inputs, kernel_initiali

    model.add(layers.Dense(2000, activation='relu'))

    model.add(layers.Dense(1))

    model.compile(loss='mean_squared_error', optimizer=optimizerinput)
    return model
```

In [20]:
```
from keras.wrappers.scikit_learn import KerasRegressor
from keras.models import Sequential
from tensorflow.keras import layers

#Hyperparameter Tuning to Tune Batch Size and Number of Epochs

from sklearn.model_selection import GridSearchCV
#https://scikit-learn.org/stable/modules/generated/sklearn.model_selecti

# create model
model = KerasRegressor(build_fn=get_nn_simplemodel, n_inputs=len(X.colum
#model = KerasClassifier(build_fn=create_model, verbose=0)

# define the grid search parameters
batch_size = [5, 10, 20, 30, 40]
epochs = [10, 50, 100]
param_grid = dict(batch_size=batch_size, epochs=epochs)
grid = GridSearchCV(estimator=model, param_grid=param_grid, n_jobs=-1, c
grid_result = grid.fit(X, y)
# summarize results

print(grid_result)
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_p
means = grid_result.cv_results_['mean_test_score']
stds = grid_result.cv_results_['std_test_score']
params = grid_result.cv_results_['params']
for mean, stdev, param in zip(means, stds, params):
    print("%f (%f) with: %r" % (mean, stdev, param))
```

```
Epoch 1/50
114/114 [==============================] - 0s 1ms/step - loss: 0.1086
Epoch 2/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0745
Epoch 3/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0578
Epoch 4/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0411
Epoch 5/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0363
Epoch 6/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0319
Epoch 7/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0299
Epoch 8/50
```

```
114/114 [==============================] - 0s 1ms/step - loss: 0.0246
Epoch 9/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0241
Epoch 10/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0306
Epoch 11/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0214
Epoch 12/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0192
Epoch 13/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0197
Epoch 14/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0194
Epoch 15/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0293
Epoch 16/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0191
Epoch 17/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0189
Epoch 18/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0206
Epoch 19/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0182
Epoch 20/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0191
Epoch 21/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0189
Epoch 22/50
114/114 [==============================] - 0s 2ms/step - loss: 0.0166
Epoch 23/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0192
Epoch 24/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0198
Epoch 25/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0189
Epoch 26/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0229
Epoch 27/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0230
Epoch 28/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0147
Epoch 29/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0159
Epoch 30/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0150
Epoch 31/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0165
Epoch 32/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0160
Epoch 33/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0185
Epoch 34/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0151
Epoch 35/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0259
Epoch 36/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0187
Epoch 37/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0135
Epoch 38/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0142
Epoch 39/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0247
Epoch 40/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0152
Epoch 41/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0146
Epoch 42/50
```

```
114/114 [==============================] - 0s 1ms/step - loss: 0.0149
Epoch 43/50
114/114 [==============================] - 0s 2ms/step - loss: 0.0140
Epoch 44/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0127
Epoch 45/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0163
Epoch 46/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0166
Epoch 47/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0140
Epoch 48/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0133
Epoch 49/50
114/114 [==============================] - 0s 1ms/step - loss: 0.0168
Epoch 50/50
114/114 [==============================] - 0s 2ms/step - loss: 0.0146
GridSearchCV(cv=3,
             estimator=<tensorflow.python.keras.wrappers.scikit_learn.Ker
asRegressor object at 0x0000016D0D05BA60>,
             n_jobs=-1,
             param_grid={'batch_size': [5, 10, 20, 30, 40],
                         'epochs': [10, 50, 100]})
Best: -0.023331 using {'batch_size': 5, 'epochs': 50}
-0.037746 (0.011950) with: {'batch_size': 5, 'epochs': 10}
-0.023331 (0.012707) with: {'batch_size': 5, 'epochs': 50}
-0.025792 (0.014083) with: {'batch_size': 5, 'epochs': 100}
-0.050562 (0.017155) with: {'batch_size': 10, 'epochs': 10}
-0.025862 (0.003204) with: {'batch_size': 10, 'epochs': 50}
-0.029090 (0.005693) with: {'batch_size': 10, 'epochs': 100}
-0.050995 (0.005893) with: {'batch_size': 20, 'epochs': 10}
-0.026017 (0.006634) with: {'batch_size': 20, 'epochs': 50}
-0.024380 (0.010593) with: {'batch_size': 20, 'epochs': 100}
-0.062723 (0.020652) with: {'batch_size': 30, 'epochs': 10}
-0.026098 (0.012060) with: {'batch_size': 30, 'epochs': 50}
-0.023892 (0.010491) with: {'batch_size': 30, 'epochs': 100}
-0.072307 (0.019946) with: {'batch_size': 40, 'epochs': 10}
-0.029889 (0.005592) with: {'batch_size': 40, 'epochs': 50}
-0.026316 (0.003739) with: {'batch_size': 40, 'epochs': 100}
```

In [21]:

```
#Hyperparameter Tuning to Tune Optimization Algorithm

from sklearn.model_selection import GridSearchCV

# create model
model = KerasRegressor(build_fn=get_nn_simplemodel, n_inputs=len(X.colum

# define the grid search parameters
optimizer = ['RMSprop', 'Adagrad', 'Adadelta', 'Adam', 'Adamax', 'Nadam'
param_grid = dict(optimizerinput=optimizer)
grid = GridSearchCV(estimator=model, param_grid=param_grid, n_jobs=-1, c
grid_result = grid.fit(X, y)
# summarize results
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_p
means = grid_result.cv_results_['mean_test_score']
stds = grid_result.cv_results_['std_test_score']
params = grid_result.cv_results_['params']
for mean, stdev, param in zip(means, stds, params):
    print("%f (%f) with: %r" % (mean, stdev, param))
```

```
Epoch 1/200
114/114 [==============================] - 0s 1ms/step - loss: 0.1094
Epoch 2/200
114/114 [==============================] - 0s 851us/step - loss: 0.0699
Epoch 3/200
```

```
114/114 [==============================] - 0s 817us/step - loss: 0.0646
Epoch 4/200
114/114 [==============================] - 0s 839us/step - loss: 0.0584
Epoch 5/200
114/114 [==============================] - 0s 923us/step - loss: 0.0547
Epoch 6/200
114/114 [==============================] - 0s 818us/step - loss: 0.0499
Epoch 7/200
114/114 [==============================] - 0s 819us/step - loss: 0.0458
Epoch 8/200
114/114 [==============================] - 0s 1ms/step - loss: 0.0427
Epoch 9/200
114/114 [==============================] - 0s 821us/step - loss: 0.0402
Epoch 10/200
114/114 [==============================] - 0s 943us/step - loss: 0.0377
Epoch 11/200
114/114 [==============================] - 0s 1ms/step - loss: 0.0366
Epoch 12/200
114/114 [==============================] - 0s 812us/step - loss: 0.0346
Epoch 13/200
114/114 [==============================] - 0s 808us/step - loss: 0.0334
Epoch 14/200
114/114 [==============================] - 0s 852us/step - loss: 0.0309
Epoch 15/200
114/114 [==============================] - 0s 908us/step - loss: 0.0284
Epoch 16/200
114/114 [==============================] - 0s 793us/step - loss: 0.0268
Epoch 17/200
114/114 [==============================] - 0s 866us/step - loss: 0.0261
Epoch 18/200
114/114 [==============================] - 0s 809us/step - loss: 0.0248
Epoch 19/200
114/114 [==============================] - 0s 808us/step - loss: 0.0243
Epoch 20/200
114/114 [==============================] - 0s 921us/step - loss: 0.0232
Epoch 21/200
114/114 [==============================] - 0s 834us/step - loss: 0.0225
Epoch 22/200
114/114 [==============================] - 0s 1ms/step - loss: 0.0215
Epoch 23/200
114/114 [==============================] - 0s 821us/step - loss: 0.0220
Epoch 24/200
114/114 [==============================] - 0s 819us/step - loss: 0.0200
Epoch 25/200
114/114 [==============================] - 0s 861us/step - loss: 0.0214
Epoch 26/200
114/114 [==============================] - 0s 903us/step - loss: 0.0180
Epoch 27/200
114/114 [==============================] - 0s 792us/step - loss: 0.0185
Epoch 28/200
114/114 [==============================] - 0s 842us/step - loss: 0.0176
Epoch 29/200
114/114 [==============================] - 0s 837us/step - loss: 0.0176
Epoch 30/200
114/114 [==============================] - 0s 958us/step - loss: 0.0177
Epoch 31/200
114/114 [==============================] - 0s 930us/step - loss: 0.0174
Epoch 32/200
114/114 [==============================] - 0s 1ms/step - loss: 0.0166
Epoch 33/200
114/114 [==============================] - 0s 853us/step - loss: 0.0164
Epoch 34/200
114/114 [==============================] - 0s 839us/step - loss: 0.0156
Epoch 35/200
114/114 [==============================] - 0s 849us/step - loss: 0.0159
Epoch 36/200
114/114 [==============================] - 0s 827us/step - loss: 0.0145
Epoch 37/200
```

```
114/114 [==============================] - 0s 872us/step - loss: 0.0153
Epoch 38/200
114/114 [==============================] - 0s 824us/step - loss: 0.0149
Epoch 39/200
114/114 [==============================] - 0s 846us/step - loss: 0.0140
Epoch 40/200
114/114 [==============================] - 0s 871us/step - loss: 0.0150
Epoch 41/200
114/114 [==============================] - 0s 968us/step - loss: 0.0140
Epoch 42/200
114/114 [==============================] - 0s 929us/step - loss: 0.0138
Epoch 43/200
114/114 [==============================] - 0s 985us/step - loss: 0.0143
Epoch 44/200
114/114 [==============================] - 0s 811us/step - loss: 0.0133
Epoch 45/200
114/114 [==============================] - 0s 791us/step - loss: 0.0126
Epoch 46/200
114/114 [==============================] - 0s 896us/step - loss: 0.0131
Epoch 47/200
114/114 [==============================] - 0s 849us/step - loss: 0.0145
Epoch 48/200
114/114 [==============================] - 0s 823us/step - loss: 0.0132
Epoch 49/200
114/114 [==============================] - 0s 834us/step - loss: 0.0134
Epoch 50/200
114/114 [==============================] - 0s 849us/step - loss: 0.0125
Epoch 51/200
114/114 [==============================] - 0s 912us/step - loss: 0.0121
Epoch 52/200
114/114 [==============================] - 0s 856us/step - loss: 0.0117
Epoch 53/200
114/114 [==============================] - 0s 1ms/step - loss: 0.0121
Epoch 54/200
114/114 [==============================] - 0s 842us/step - loss: 0.0119
Epoch 55/200
114/114 [==============================] - 0s 788us/step - loss: 0.0118
Epoch 56/200
114/114 [==============================] - 0s 871us/step - loss: 0.0118
Epoch 57/200
114/114 [==============================] - 0s 851us/step - loss: 0.0129
Epoch 58/200
114/114 [==============================] - 0s 817us/step - loss: 0.0109
Epoch 59/200
114/114 [==============================] - 0s 838us/step - loss: 0.0130
Epoch 60/200
114/114 [==============================] - 0s 860us/step - loss: 0.0113
Epoch 61/200
114/114 [==============================] - 0s 858us/step - loss: 0.0147
Epoch 62/200
114/114 [==============================] - 0s 841us/step - loss: 0.0118
Epoch 63/200
114/114 [==============================] - 0s 887us/step - loss: 0.0116
Epoch 64/200
114/114 [==============================] - 0s 901us/step - loss: 0.0103
Epoch 65/200
114/114 [==============================] - 0s 851us/step - loss: 0.0118
Epoch 66/200
114/114 [==============================] - 0s 840us/step - loss: 0.0113
Epoch 67/200
114/114 [==============================] - 0s 926us/step - loss: 0.0115
Epoch 68/200
114/114 [==============================] - 0s 793us/step - loss: 0.0105
Epoch 69/200
114/114 [==============================] - 0s 799us/step - loss: 0.0124
Epoch 70/200
114/114 [==============================] - 0s 880us/step - loss: 0.0104
Epoch 71/200
```

```
114/114 [==============================] - 0s 810us/step - loss: 0.0103
Epoch 72/200
114/114 [==============================] - 0s 876us/step - loss: 0.0104
Epoch 73/200
114/114 [==============================] - 0s 808us/step - loss: 0.0098
Epoch 74/200
114/114 [==============================] - 0s 1ms/step - loss: 0.0103
Epoch 75/200
114/114 [==============================] - 0s 818us/step - loss: 0.0105
Epoch 76/200
114/114 [==============================] - 0s 828us/step - loss: 0.0105
Epoch 77/200
114/114 [==============================] - 0s 857us/step - loss: 0.0101
Epoch 78/200
114/114 [==============================] - 0s 799us/step - loss: 0.0110
Epoch 79/200
114/114 [==============================] - 0s 795us/step - loss: 0.0098
Epoch 80/200
114/114 [==============================] - 0s 834us/step - loss: 0.0093
Epoch 81/200
114/114 [==============================] - 0s 797us/step - loss: 0.0114
Epoch 82/200
114/114 [==============================] - 0s 801us/step - loss: 0.0101
Epoch 83/200
114/114 [==============================] - 0s 911us/step - loss: 0.0088
Epoch 84/200
114/114 [==============================] - 0s 800us/step - loss: 0.0108
Epoch 85/200
114/114 [==============================] - 0s 978us/step - loss: 0.0098
Epoch 86/200
114/114 [==============================] - 0s 791us/step - loss: 0.0087
Epoch 87/200
114/114 [==============================] - 0s 805us/step - loss: 0.0085
Epoch 88/200
114/114 [==============================] - 0s 827us/step - loss: 0.0088
Epoch 89/200
114/114 [==============================] - 0s 827us/step - loss: 0.0105
Epoch 90/200
114/114 [==============================] - 0s 804us/step - loss: 0.0086
Epoch 91/200
114/114 [==============================] - 0s 810us/step - loss: 0.0082
Epoch 92/200
114/114 [==============================] - 0s 794us/step - loss: 0.0093
Epoch 93/200
114/114 [==============================] - 0s 861us/step - loss: 0.0109
Epoch 94/200
114/114 [==============================] - 0s 839us/step - loss: 0.0096
Epoch 95/200
114/114 [==============================] - 0s 791us/step - loss: 0.0091
Epoch 96/200
114/114 [==============================] - 0s 1ms/step - loss: 0.0098
Epoch 97/200
114/114 [==============================] - 0s 805us/step - loss: 0.0087
Epoch 98/200
114/114 [==============================] - 0s 791us/step - loss: 0.0076
Epoch 99/200
114/114 [==============================] - 0s 830us/step - loss: 0.0091
Epoch 100/200
114/114 [==============================] - 0s 825us/step - loss: 0.0075
Epoch 101/200
114/114 [==============================] - 0s 815us/step - loss: 0.0099
Epoch 102/200
114/114 [==============================] - 0s 827us/step - loss: 0.0090
Epoch 103/200
114/114 [==============================] - 0s 799us/step - loss: 0.0077
Epoch 104/200
114/114 [==============================] - 0s 830us/step - loss: 0.0076
Epoch 105/200
```

```
114/114 [==============================] - 0s 787us/step - loss: 0.0075
Epoch 106/200
114/114 [==============================] - 0s 787us/step - loss: 0.0073
Epoch 107/200
114/114 [==============================] - 0s 981us/step - loss: 0.0073
Epoch 108/200
114/114 [==============================] - 0s 825us/step - loss: 0.0076
Epoch 109/200
114/114 [==============================] - 0s 788us/step - loss: 0.0084
Epoch 110/200
114/114 [==============================] - 0s 832us/step - loss: 0.0074
Epoch 111/200
114/114 [==============================] - 0s 799us/step - loss: 0.0067
Epoch 112/200
114/114 [==============================] - 0s 778us/step - loss: 0.0077
Epoch 113/200
114/114 [==============================] - 0s 804us/step - loss: 0.0064
Epoch 114/200
114/114 [==============================] - 0s 778us/step - loss: 0.0070
Epoch 115/200
114/114 [==============================] - 0s 819us/step - loss: 0.0072
Epoch 116/200
114/114 [==============================] - 0s 817us/step - loss: 0.0069
Epoch 117/200
114/114 [==============================] - 0s 835us/step - loss: 0.0093
Epoch 118/200
114/114 [==============================] - 0s 998us/step - loss: 0.0086
Epoch 119/200
114/114 [==============================] - 0s 785us/step - loss: 0.0072
Epoch 120/200
114/114 [==============================] - 0s 798us/step - loss: 0.0079
Epoch 121/200
114/114 [==============================] - 0s 859us/step - loss: 0.0058
Epoch 122/200
114/114 [==============================] - 0s 828us/step - loss: 0.0055
Epoch 123/200
114/114 [==============================] - 0s 779us/step - loss: 0.0070
Epoch 124/200
114/114 [==============================] - 0s 823us/step - loss: 0.0071
Epoch 125/200
114/114 [==============================] - 0s 816us/step - loss: 0.0062
Epoch 126/200
114/114 [==============================] - 0s 842us/step - loss: 0.0070
Epoch 127/200
114/114 [==============================] - 0s 815us/step - loss: 0.0057
Epoch 128/200
114/114 [==============================] - 0s 788us/step - loss: 0.0060
Epoch 129/200
114/114 [==============================] - 0s 1ms/step - loss: 0.0068
Epoch 130/200
114/114 [==============================] - 0s 826us/step - loss: 0.0072
Epoch 131/200
114/114 [==============================] - 0s 787us/step - loss: 0.0059
Epoch 132/200
114/114 [==============================] - 0s 885us/step - loss: 0.0058
Epoch 133/200
114/114 [==============================] - 0s 797us/step - loss: 0.0060
Epoch 134/200
114/114 [==============================] - 0s 807us/step - loss: 0.0065
Epoch 135/200
114/114 [==============================] - 0s 793us/step - loss: 0.0064
Epoch 136/200
114/114 [==============================] - 0s 798us/step - loss: 0.0071
Epoch 137/200
114/114 [==============================] - 0s 827us/step - loss: 0.0059
Epoch 138/200
114/114 [==============================] - 0s 785us/step - loss: 0.0070
Epoch 139/200
```

```
114/114 [==============================] - 0s 771us/step - loss: 0.0063
Epoch 140/200
114/114 [==============================] - 0s 1ms/step - loss: 0.0061
Epoch 141/200
114/114 [==============================] - 0s 814us/step - loss: 0.0050
Epoch 142/200
114/114 [==============================] - 0s 843us/step - loss: 0.0055
Epoch 143/200
114/114 [==============================] - 0s 859us/step - loss: 0.0063
Epoch 144/200
114/114 [==============================] - 0s 834us/step - loss: 0.0056
Epoch 145/200
114/114 [==============================] - 0s 830us/step - loss: 0.0059
Epoch 146/200
114/114 [==============================] - 0s 839us/step - loss: 0.0060
Epoch 147/200
114/114 [==============================] - 0s 790us/step - loss: 0.0061
Epoch 148/200
114/114 [==============================] - 0s 839us/step - loss: 0.0059
Epoch 149/200
114/114 [==============================] - 0s 843us/step - loss: 0.0051
Epoch 150/200
114/114 [==============================] - 0s 846us/step - loss: 0.0061
Epoch 151/200
114/114 [==============================] - 0s 960us/step - loss: 0.0062
Epoch 152/200
114/114 [==============================] - 0s 855us/step - loss: 0.0049
Epoch 153/200
114/114 [==============================] - 0s 917us/step - loss: 0.0055
Epoch 154/200
114/114 [==============================] - 0s 813us/step - loss: 0.0069
Epoch 155/200
114/114 [==============================] - 0s 813us/step - loss: 0.0052
Epoch 156/200
114/114 [==============================] - 0s 798us/step - loss: 0.0047
Epoch 157/200
114/114 [==============================] - 0s 815us/step - loss: 0.0060
Epoch 158/200
114/114 [==============================] - 0s 824us/step - loss: 0.0054
Epoch 159/200
114/114 [==============================] - 0s 899us/step - loss: 0.0047
Epoch 160/200
114/114 [==============================] - 0s 804us/step - loss: 0.0047
Epoch 161/200
114/114 [==============================] - 0s 816us/step - loss: 0.0057
Epoch 162/200
114/114 [==============================] - 0s 984us/step - loss: 0.0046
Epoch 163/200
114/114 [==============================] - 0s 870us/step - loss: 0.0056
Epoch 164/200
114/114 [==============================] - 0s 846us/step - loss: 0.0047
Epoch 165/200
114/114 [==============================] - 0s 831us/step - loss: 0.0054
Epoch 166/200
114/114 [==============================] - 0s 815us/step - loss: 0.0068
Epoch 167/200
114/114 [==============================] - 0s 883us/step - loss: 0.0049
Epoch 168/200
114/114 [==============================] - 0s 796us/step - loss: 0.0050
Epoch 169/200
114/114 [==============================] - 0s 866us/step - loss: 0.0050
Epoch 170/200
114/114 [==============================] - 0s 1ms/step - loss: 0.0054
Epoch 171/200
114/114 [==============================] - 0s 817us/step - loss: 0.0047
Epoch 172/200
114/114 [==============================] - 0s 1ms/step - loss: 0.0042
Epoch 173/200
```

```
114/114 [==============================] - 0s 859us/step - loss: 0.0050
Epoch 174/200
114/114 [==============================] - 0s 914us/step - loss: 0.0052
Epoch 175/200
114/114 [==============================] - 0s 824us/step - loss: 0.0044
Epoch 176/200
114/114 [==============================] - 0s 820us/step - loss: 0.0062
Epoch 177/200
114/114 [==============================] - 0s 789us/step - loss: 0.0050
Epoch 178/200
114/114 [==============================] - 0s 829us/step - loss: 0.0045
Epoch 179/200
114/114 [==============================] - 0s 840us/step - loss: 0.0053
Epoch 180/200
114/114 [==============================] - 0s 966us/step - loss: 0.0045
Epoch 181/200
114/114 [==============================] - 0s 830us/step - loss: 0.0055
Epoch 182/200
114/114 [==============================] - 0s 914us/step - loss: 0.0036
Epoch 183/200
114/114 [==============================] - 0s 915us/step - loss: 0.0039
Epoch 184/200
114/114 [==============================] - 0s 853us/step - loss: 0.0052
Epoch 185/200
114/114 [==============================] - 0s 816us/step - loss: 0.0042
Epoch 186/200
114/114 [==============================] - 0s 789us/step - loss: 0.0056
Epoch 187/200
114/114 [==============================] - 0s 813us/step - loss: 0.0041
Epoch 188/200
114/114 [==============================] - 0s 807us/step - loss: 0.0038
Epoch 189/200
114/114 [==============================] - 0s 828us/step - loss: 0.0046
Epoch 190/200
114/114 [==============================] - 0s 854us/step - loss: 0.0048
Epoch 191/200
114/114 [==============================] - 0s 806us/step - loss: 0.0042
Epoch 192/200
114/114 [==============================] - 0s 810us/step - loss: 0.0045
Epoch 193/200
114/114 [==============================] - 0s 866us/step - loss: 0.0041
Epoch 194/200
114/114 [==============================] - 0s 878us/step - loss: 0.0040
Epoch 195/200
114/114 [==============================] - 0s 947us/step - loss: 0.0034
Epoch 196/200
114/114 [==============================] - 0s 814us/step - loss: 0.0051
Epoch 197/200
114/114 [==============================] - 0s 786us/step - loss: 0.0034
Epoch 198/200
114/114 [==============================] - 0s 813us/step - loss: 0.0035
Epoch 199/200
114/114 [==============================] - 0s 863us/step - loss: 0.0046
Epoch 200/200
114/114 [==============================] - 0s 834us/step - loss: 0.0035
Best: -0.023221 using {'optimizerinput': 'Adamax'}
-0.034766 (0.008045) with: {'optimizerinput': 'RMSprop'}
-0.062387 (0.018451) with: {'optimizerinput': 'Adagrad'}
-0.130494 (0.025806) with: {'optimizerinput': 'Adadelta'}
-0.029354 (0.012870) with: {'optimizerinput': 'Adam'}
-0.023221 (0.009218) with: {'optimizerinput': 'Adamax'}
-0.047123 (0.015469) with: {'optimizerinput': 'Nadam'}
```

In [22]:

```python
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold

#Get error using cross validation
```

```python
# evaluate model
estimator = KerasRegressor(build_fn=get_nn_simplemodel, n_inputs=len(X.c

kfold = KFold(n_splits=5)
results = cross_val_score(estimator, X, y, cv=kfold)
print("Baseline: %.2f (%.2f) MSE" % (results.mean(), results.std()))
```

```
Baseline: -0.03 (0.01) MSE
```

In [23]:
```python
model=get_nn_simplemodel(len(X.columns),1, optimizerinput = 'Adam');
history = model.fit(train_X, train_y, verbose=1, epochs=100, batch_size=
```

```
Epoch 1/100
46/46 [==============================] - 0s 4ms/step - loss: 0.1442 - val
_loss: 0.0814
Epoch 2/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0737 - val
_loss: 0.0637
Epoch 3/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0630 - val
_loss: 0.0541
Epoch 4/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0564 - val
_loss: 0.0614
Epoch 5/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0589 - val
_loss: 0.0461
Epoch 6/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0447 - val
_loss: 0.0422
Epoch 7/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0429 - val
_loss: 0.0405
Epoch 8/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0350 - val
_loss: 0.0430
Epoch 9/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0354 - val
_loss: 0.0441
Epoch 10/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0349 - val
_loss: 0.0304
Epoch 11/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0294 - val
_loss: 0.0319
Epoch 12/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0242 - val
_loss: 0.0238
Epoch 13/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0214 - val
_loss: 0.0270
Epoch 14/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0216 - val
_loss: 0.0229
Epoch 15/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0219 - val
_loss: 0.0260
Epoch 16/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0219 - val
_loss: 0.0341
Epoch 17/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0292 - val
_loss: 0.0240
Epoch 18/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0178 - val
_loss: 0.0201
Epoch 19/100
```

```
46/46 [==============================] - 0s 2ms/step - loss: 0.0170 - val
_loss: 0.0246
Epoch 20/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0188 - val
_loss: 0.0250
Epoch 21/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0148 - val
_loss: 0.0298
Epoch 22/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0123 - val
_loss: 0.0253
Epoch 23/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0145 - val
_loss: 0.0230
Epoch 24/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0198 - val
_loss: 0.0290
Epoch 25/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0123 - val
_loss: 0.0240
Epoch 26/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0231 - val
_loss: 0.0338
Epoch 27/100
46/46 [==============================] - 0s 3ms/step - loss: 0.0173 - val
_loss: 0.0230
Epoch 28/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0139 - val
_loss: 0.0251
Epoch 29/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0125 - val
_loss: 0.0210
Epoch 30/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0210 - val
_loss: 0.0288
Epoch 31/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0137 - val
_loss: 0.0308
Epoch 32/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0131 - val
_loss: 0.0243
Epoch 33/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0115 - val
_loss: 0.0351
Epoch 34/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0146 - val
_loss: 0.0377
Epoch 35/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0124 - val
_loss: 0.0265
Epoch 36/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0105 - val
_loss: 0.0284
Epoch 37/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0169 - val
_loss: 0.0264
Epoch 38/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0122 - val
_loss: 0.0277
Epoch 39/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0139 - val
_loss: 0.0302
Epoch 40/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0167 - val
_loss: 0.0263
Epoch 41/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0112 - val
_loss: 0.0251
```

```
Epoch 42/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0103 - val
_loss: 0.0248
Epoch 43/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0103 - val
_loss: 0.0258
Epoch 44/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0144 - val
_loss: 0.0367
Epoch 45/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0136 - val
_loss: 0.0242
Epoch 46/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0083 - val
_loss: 0.0317
Epoch 47/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0124 - val
_loss: 0.0258
Epoch 48/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0111 - val
_loss: 0.0282
Epoch 49/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0152 - val
_loss: 0.0304
Epoch 50/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0143 - val
_loss: 0.0272
Epoch 51/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0184 - val
_loss: 0.0407
Epoch 52/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0128 - val
_loss: 0.0256
Epoch 53/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0078 - val
_loss: 0.0237
Epoch 54/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0090 - val
_loss: 0.0318
Epoch 55/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0104 - val
_loss: 0.0285
Epoch 56/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0115 - val
_loss: 0.0276
Epoch 57/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0100 - val
_loss: 0.0238
Epoch 58/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0096 - val
_loss: 0.0272
Epoch 59/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0077 - val
_loss: 0.0256
Epoch 60/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0077 - val
_loss: 0.0268
Epoch 61/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0086 - val
_loss: 0.0281
Epoch 62/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0095 - val
_loss: 0.0460
Epoch 63/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0093 - val
_loss: 0.0299
Epoch 64/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0069 - val
```

```
_loss: 0.0315
Epoch 65/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0095 - val
_loss: 0.0315
Epoch 66/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0108 - val
_loss: 0.0340
Epoch 67/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0070 - val
_loss: 0.0295
Epoch 68/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0096 - val
_loss: 0.0313
Epoch 69/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0072 - val
_loss: 0.0317
Epoch 70/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0095 - val
_loss: 0.0291
Epoch 71/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0084 - val
_loss: 0.0271
Epoch 72/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0058 - val
_loss: 0.0414
Epoch 73/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0077 - val
_loss: 0.0290
Epoch 74/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0115 - val
_loss: 0.0320
Epoch 75/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0064 - val
_loss: 0.0332
Epoch 76/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0072 - val
_loss: 0.0352
Epoch 77/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0107 - val
_loss: 0.0345
Epoch 78/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0082 - val
_loss: 0.0321
Epoch 79/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0073 - val
_loss: 0.0279
Epoch 80/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0070 - val
_loss: 0.0280
Epoch 81/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0058 - val
_loss: 0.0294
Epoch 82/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0157 - val
_loss: 0.0391
Epoch 83/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0076 - val
_loss: 0.0310
Epoch 84/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0058 - val
_loss: 0.0394
Epoch 85/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0063 - val
_loss: 0.0350
Epoch 86/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0062 - val
_loss: 0.0347
Epoch 87/100
```

```
46/46 [==============================] - 0s 1ms/step - loss: 0.0041 - val
_loss: 0.0298
Epoch 88/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0056 - val
_loss: 0.0265
Epoch 89/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0054 - val
_loss: 0.0390
Epoch 90/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0056 - val
_loss: 0.0287
Epoch 91/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0062 - val
_loss: 0.0303
Epoch 92/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0063 - val
_loss: 0.0345
Epoch 93/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0071 - val
_loss: 0.0360
Epoch 94/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0047 - val
_loss: 0.0340
Epoch 95/100
46/46 [==============================] - 0s 1ms/step - loss: 0.0046 - val
_loss: 0.0386
Epoch 96/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0072 - val
_loss: 0.0306
Epoch 97/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0073 - val
_loss: 0.0445
Epoch 98/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0051 - val
_loss: 0.0296
Epoch 99/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0066 - val
_loss: 0.0282
Epoch 100/100
46/46 [==============================] - 0s 2ms/step - loss: 0.0057 - val
_loss: 0.0501
```

In [24]:

```python
from datetime import datetime

# Code to create graph for train vs validation error for different epoch

loss_train = history.history['loss']
loss_val = history.history['val_loss']

diff_in_loss=abs(np.subtract(loss_val,loss_train));

df = pd.DataFrame({'loss_train':loss_train, 'loss_val':loss_val, 'diff_i

print(df.head(10))

epochs = range(1,len(loss_train)+1)
plt.plot(epochs, loss_train, 'g', label='Training loss')
plt.plot(epochs, loss_val, 'b', label='validation loss')
plt.plot(epochs, diff_in_loss, 'r', label='diff in loss')

plt.title('Training and Validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

timeStr=datetime.now().strftime("%Y%m%d-%H%M%S");
```

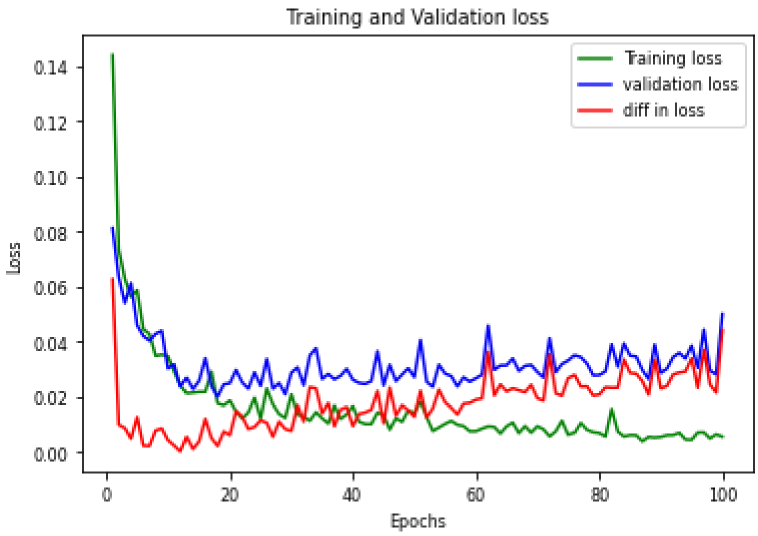
```
fileName = 'train_vs_Validation_loss_'+timeStr

plt.savefig(fileName+'.png',format='png', dpi=2000)

df.to_excel("neural_network_train_vs_val_loss"+timeStr+".xlsx", sheet_nam

plt.show()
```

```
   loss_train  loss_val  diff_in_loss
0    0.144155  0.081368      0.062787
1    0.073730  0.063680      0.010050
2    0.063003  0.054149      0.008854
3    0.056380  0.061395      0.005015
4    0.058851  0.046058      0.012793
5    0.044713  0.042212      0.002500
6    0.042888  0.040501      0.002387
7    0.035001  0.042954      0.007954
8    0.035377  0.044071      0.008694
9    0.034865  0.030361      0.004504
```



In [25]:
```
#code to print actual, predicted, diff in values

predicted=model.predict(val_X);
actual=val_y;
```

In [26]:
```
y_flat=actual.values.flatten();

predicted_flat=predicted.flatten()

residue=abs(np.subtract(y_flat,predicted_flat))
```

In [27]:
```
#Plot
xasix = range(1,len(y_flat)+1)

plt.plot(xasix, y_flat, 'g', label='Original value')
plt.plot(xasix, predicted_flat, 'b', label='predicted valye')
plt.plot(xasix, residue, 'r', label='diff in value')

plt.title('Original vs Predicted values')
plt.xlabel('id')
plt.ylabel('value')
plt.legend()

timeStr=datetime.now().strftime("%Y%m%d-%H%M%S");
fileName = 'Original_vs_Predicted_y'+timeStr
```

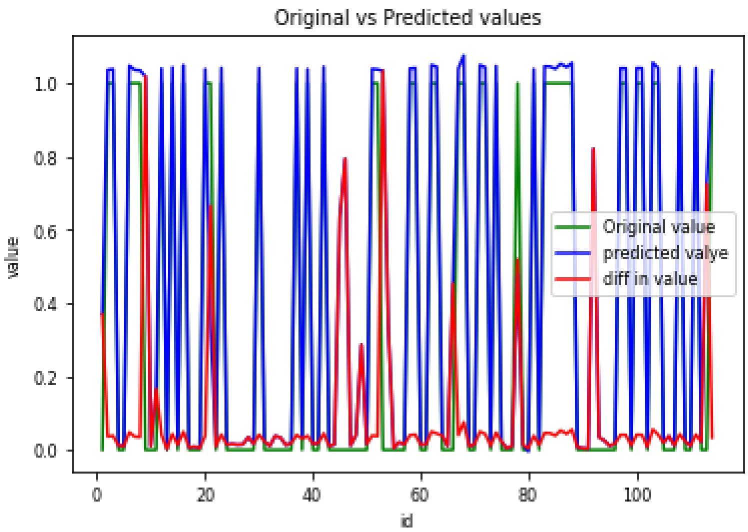
```
plt.savefig(fileName+'.png',format='png', dpi=2000)

plt.show()

#Write to file
df = pd.DataFrame({'y1':y_flat, 'pred':predicted_flat, 'residue':residue

df.to_excel("neural_network_output"+timeStr+".xlsx", sheet_name='Sheet_n

#Data frame to include all X,y, predicted y and difference between actua
masterdf = X.merge(df,left_index=True, right_index=True)
masterdf.to_excel("master_analysis_"+timeStr+".xlsx", sheet_name='Sheet_
```




```
In [ ]:
```