```python
# -*- coding: utf-8 -*-
"""
Created on Wed Apr 29 20:55:55 2020

@author: Debanjan Parbat

Support Vector Regression Model
"""
#Import the libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn import metrics

#import dataset

dataset = pd.read_csv(r'C:\Users\Debanjan\Desktop\novel-corona-virus-2019-dataset (3)\covid19-in-ir

# create feauture and label column

X = np.arange(60)
X = X.reshape(-1,1)

y = dataset.iloc[:,-1].values.astype(float)

# create the daily number of cases series

y = np.diff(y)
y = y.reshape(-1,1)

# create train and test dataset

x_train, x_test, y_train, y_test = train_test_split(X,y,test_size=0.40,random_state=1)

#scaling the datest by standardization technique

from sklearn.preprocessing import StandardScaler
scx = StandardScaler()
scy = StandardScaler()
x_train = scx.fit_transform(x_train)
x_test = scx.transform(x_test)
y_train = scy.fit_transform(y_train)
y_test = scy.transform(y_test)

#Fit the SVR model

from sklearn.svm import SVR
regressor = SVR(kernel='rbf', epsilon=0.1,)
regressor.fit(x_train,y_train)
y_pred = regressor.predict(x_test)

# Caluclate the model performance parameters

mse = metrics.mean_squared_error(y_test,y_pred)
rmse = np.sqrt(mse)
```

```
print(mse)
print(rmse)
print(regressor.score(x_test,y_test))

y_pred = np.array(scy.inverse_transform(y_pred))
y_pred = y_pred.reshape(-1,1)
x_test = np.array(scx.inverse_transform(x_test))

#plot the regression fit with original data

plt.scatter(X, y, color = 'magenta', label = 'Original Data')
plt.scatter(x_test, y_pred, color = 'green', label = 'Test Data')
plt.title('Covid19 (Support Vector Regression Model)')
plt.xlabel('Days')
plt.ylabel('Daily New Cases')
plt.legend()
plt.show()

#Predict the future values by entering number of days since onset

y_pred = scy.inverse_transform(regressor.predict(scx.transform([[100]])))
print(y_pred)
```