In [1]:

```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
from sklearn . tree import DecisionTreeRegressor
from sklearn . linear_model import LinearRegression
from sklearn . model_selection import train_test_split
import matplotlib . pyplot as plt
```

In [3]:

```
plt.style.use ('bmh')
```

In [4]:

```
df = pd.read_csv ("C:/Users/Vinod Kumar/Downloads/NFLX.csv")
print ( df )
```

```
            Date        Open        High         Low       Close   Adj Close
\
0     2010-01-04    7.931429    7.961429    7.565714    7.640000    7.640000
1     2010-01-05    7.652857    7.657143    7.258571    7.358571    7.358571
2     2010-01-06    7.361429    7.672857    7.197143    7.617143    7.617143
3     2010-01-07    7.731429    7.757143    7.462857    7.485714    7.485714
4     2010-01-08    7.498571    7.742857    7.465714    7.614286    7.614286
...          ...         ...         ...         ...         ...         ...
3114  2022-05-17  189.169998  191.399994  185.169998  190.559998  190.559998
3115  2022-05-18  186.720001  187.699997  176.270004  177.190002  177.190002
3116  2022-05-19  178.050003  186.300003  175.710007  183.479996  183.479996
3117  2022-05-20  185.869995  190.190002  179.770004  186.350006  186.350006
3118  2022-05-23  186.149994  187.660004  177.889999  187.440002  187.440002

        Volume
0     17239600
1     23753100
2     23290400
3      9955400
4      8180900
...        ...
3114   9876700
3115   9665600
3116  10448500
3117  10422600
3118   9558000

[3119 rows x 7 columns]
```

In [5]:

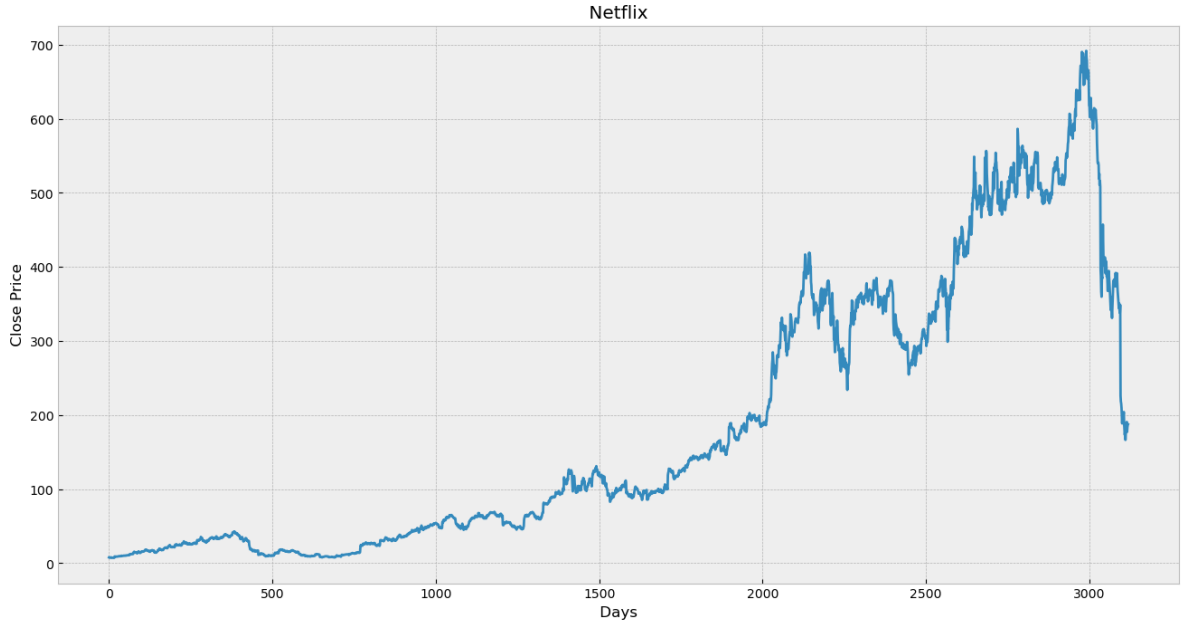
```
df.shape
```

Out[5]:

```
(3119, 7)
```

In [6]:

```
plt . figure ( figsize =(16 ,8) )
plt . title (" Netflix ")
plt . xlabel (" Days ")
plt . ylabel (" Close Price ")
plt . plot ( df ['Close'])
plt . show ()
```



In [7]:

```
df = df [[ 'Close']]
df . head (3119)
```

Out[7]:

	Close
0	7.640000
1	7.358571
2	7.617143
3	7.485714
4	7.614286
...	...
3114	190.559998
3115	177.190002
3116	183.479996
3117	186.350006
3118	187.440002

3119 rows × 1 columns

In [8]:

```
future_days =25
df ['Prediction']= df [['Close']].shift(- future_days)
df.tail (26)
```

Out[8]:

	Close	Prediction
3093	337.859985	187.440002
3094	348.609985	NaN
3095	226.190002	NaN
3096	218.220001	NaN
3097	215.520004	NaN
3098	209.910004	NaN
3099	198.399994	NaN
3100	188.539993	NaN
3101	199.520004	NaN
3102	190.360001	NaN
3103	199.460007	NaN
3104	199.869995	NaN
3105	204.009995	NaN
3106	188.320007	NaN
3107	180.970001	NaN
3108	173.100006	NaN
3109	177.660004	NaN
3110	166.369995	NaN
3111	174.309998	NaN
3112	187.639999	NaN
3113	186.509995	NaN
3114	190.559998	NaN
3115	177.190002	NaN
3116	183.479996	NaN
3117	186.350006	NaN
3118	187.440002	NaN

In [9]:

```
X = np.array ( df.drop (['Prediction'] ,1) ) [: - future_days ]
print ( X )
```

```
[[  7.64    ]
 [  7.358571]
 [  7.617143]
 ...
 [350.429993]
 [341.130005]
 [337.859985]]
```

In [10]:

```
y = np . array ( df ['Prediction']) [: - future_days ]
print ( y )
```

```
[  8.875714   8.784286   9.051429 ... 183.479996 186.350006 187.440002]
```

In [11]:

```
x_train , x_test , y_train , y_test = train_test_split (X ,y , test_size =0.25)
tree = DecisionTreeRegressor () . fit ( x_train , y_train )
lr = LinearRegression () . fit ( x_train , y_train )
```

In [12]:

```
x_future = df . drop ([ 'Prediction'] ,1) [: - future_days ]
x_future = x_future . tail ( future_days )
x_future = np . array ( x_future )
x_future
```

Out[12]:

```
array([[331.01001 ],
       [343.75    ],
       [357.529999],
       [371.399994],
       [380.600006],
       [374.589996],
       [382.920013],
       [374.48999 ],
       [375.709991],
       [373.850006],
       [378.51001 ],
       [391.820007],
       [381.470001],
       [374.589996],
       [373.470001],
       [391.5     ],
       [380.149994],
       [368.350006],
       [362.149994],
       [355.880005],
       [348.      ],
       [344.100006],
       [350.429993],
       [341.130005],
       [337.859985]])
```

In [13]:

```
tree_prediction = tree . predict ( x_future )
print ( tree_prediction )
```

```
[348.609985 226.190002 218.220001 441.950012 209.910004 199.520004
 188.539993 199.520004 190.360001 315.470001 199.869995 204.009995
 188.320007 199.520004 173.100006 177.660004 166.369995 174.309998
 187.639999 186.509995 190.559998 177.190002 183.479996 325.890015
 187.440002]
```

In [14]:

```
lr_prediction = lr . predict ( x_future )
print ( lr_prediction )
```

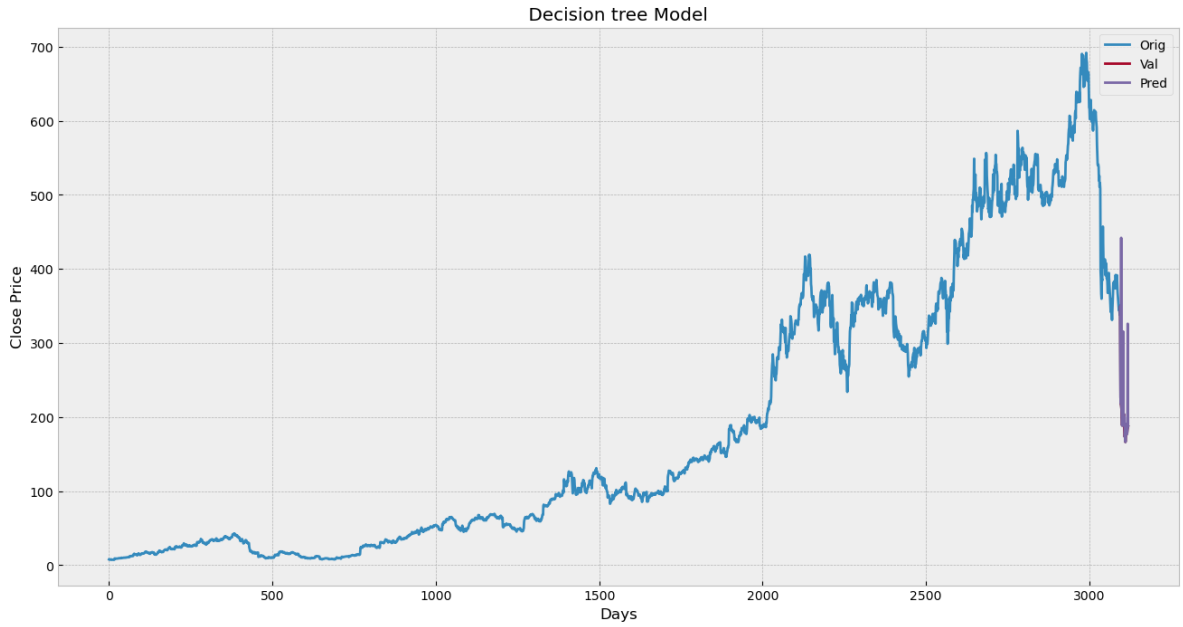
```
[329.14641216 341.59233866 355.05426875 368.60411756 377.59177557
 371.72048817 379.85823234 371.6227905  372.81463153 370.99757857
 375.5500207  388.55279734 378.44168941 371.72048817 370.62634481
 388.24017672 377.15215071 365.62452915 359.56762537 353.44235979
 345.74424048 341.93426585 348.11814456 339.03281819 335.83827655]
```

In [15]:

```
predictions = tree_prediction
valid = df [ X . shape [0]:]
valid ['Predictions']= predictions
```

In [16]:

```
plt . figure ( figsize =(16 ,8) )
plt . title (" Decision tree Model ")
plt . xlabel ('Days')
plt . ylabel ('Close Price')
plt . plot ( df ['Close'])
plt . plot ( valid [['Close','Predictions']])
plt . legend ([ 'Orig','Val','Pred'])
plt . show ()
```



In [17]:

```
predictions = lr_prediction
valid = df [ X . shape [0]:]
valid ['Predictions']= predictions
plt . figure ( figsize =(16 ,8) )
```

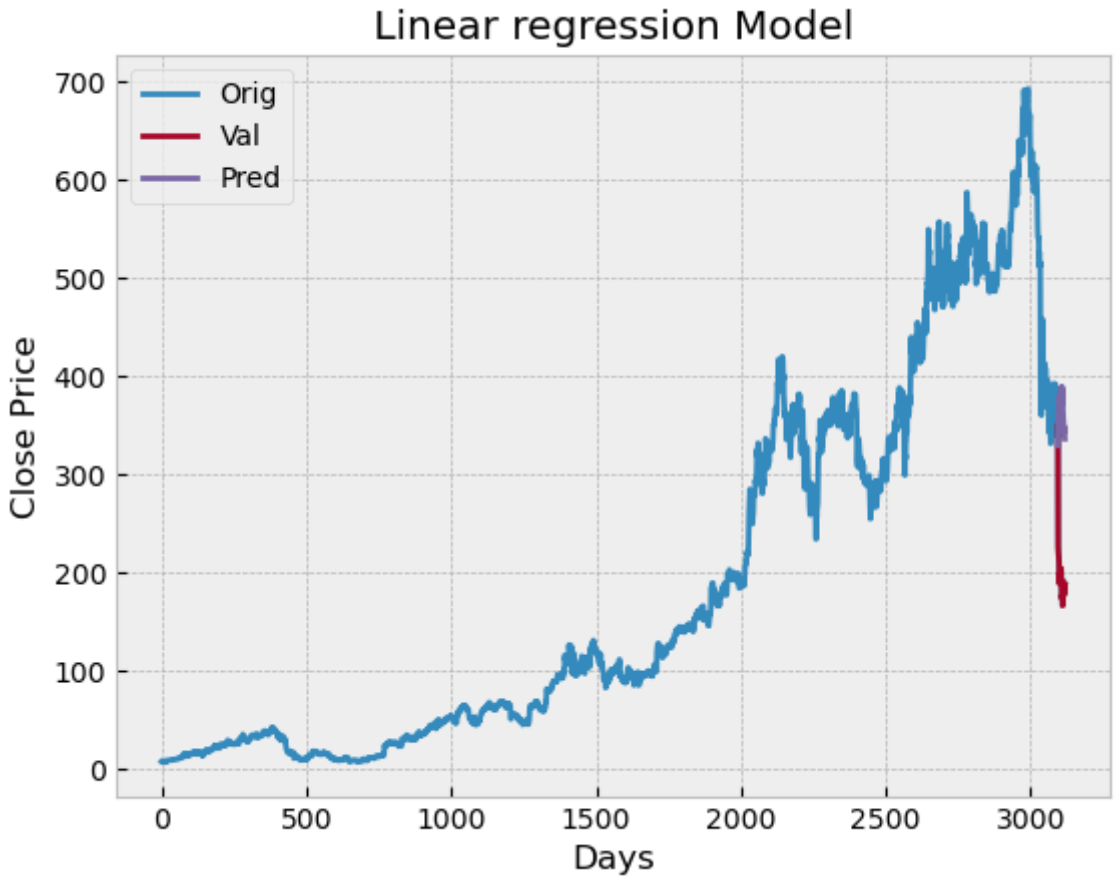
Out[17]:

```
<Figure size 1600x800 with 0 Axes>
```

```
<Figure size 1600x800 with 0 Axes>
```

In [18]:

```
plt . title (" Linear regression Model ")
plt . xlabel ('Days')
plt . ylabel ('Close Price')
plt . plot ( df ['Close'])
plt . plot ( valid [[ 'Close','Predictions']])
plt . legend (['Orig','Val','Pred'])
plt . show ()
```



In [19]:

```
df = pd . read_csv ("C:/Users/Vinod Kumar/Downloads/NFLX.csv ")
df
```

Out[19]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2010-01-04	7.931429	7.961429	7.565714	7.640000	7.640000	17239600
1	2010-01-05	7.652857	7.657143	7.258571	7.358571	7.358571	23753100
2	2010-01-06	7.361429	7.672857	7.197143	7.617143	7.617143	23290400
3	2010-01-07	7.731429	7.757143	7.462857	7.485714	7.485714	9955400
4	2010-01-08	7.498571	7.742857	7.465714	7.614286	7.614286	8180900
...	...	...	...	...	...	...	...
3114	2022-05-17	189.169998	191.399994	185.169998	190.559998	190.559998	9876700
3115	2022-05-18	186.720001	187.699997	176.270004	177.190002	177.190002	9665600
3116	2022-05-19	178.050003	186.300003	175.710007	183.479996	183.479996	10448500
3117	2022-05-20	185.869995	190.190002	179.770004	186.350006	186.350006	10422600
3118	2022-05-23	186.149994	187.660004	177.889999	187.440002	187.440002	9558000

3119 rows × 7 columns

In [20]:

```
for i in range (0 , df . shape [0] -2) :
    df . loc [ df . index [ i +2] , 'SMA_3'] = np . round ((( df . iloc [i ,1]+ df .iloc [
df
```

Out[20]:

	Date	Open	High	Low	Close	Adj Close	Volume	SMA_3
0	2010-01-04	7.931429	7.961429	7.565714	7.640000	7.640000	17239600	NaN
1	2010-01-05	7.652857	7.657143	7.258571	7.358571	7.358571	23753100	NaN
2	2010-01-06	7.361429	7.672857	7.197143	7.617143	7.617143	23290400	7.6
3	2010-01-07	7.731429	7.757143	7.462857	7.485714	7.485714	9955400	7.6
4	2010-01-08	7.498571	7.742857	7.465714	7.614286	7.614286	8180900	7.5
...	...	...	...	...	...	...	...	...
3114	2022-05-17	189.169998	191.399994	185.169998	190.559998	190.559998	9876700	186.5
3115	2022-05-18	186.720001	187.699997	176.270004	177.190002	177.190002	9665600	189.7
3116	2022-05-19	178.050003	186.300003	175.710007	183.479996	183.479996	10448500	184.6
3117	2022-05-20	185.869995	190.190002	179.770004	186.350006	186.350006	10422600	183.5
3118	2022-05-23	186.149994	187.660004	177.889999	187.440002	187.440002	9558000	183.4

3119 rows × 8 columns

In [21]:

```
for i in range (0 , df . shape [0] -3) :
    df . loc [ df . index [ i +3] , 'SMA_4'] = np . round ((( df . iloc [i ,1]+ df . iloc [
df
```

Out[21]:

	Date	Open	High	Low	Close	Adj Close	Volume	SMA_3	Sl
0	2010-01-04	7.931429	7.961429	7.565714	7.640000	7.640000	17239600	NaN	
1	2010-01-05	7.652857	7.657143	7.258571	7.358571	7.358571	23753100	NaN	
2	2010-01-06	7.361429	7.672857	7.197143	7.617143	7.617143	23290400	7.6	
3	2010-01-07	7.731429	7.757143	7.462857	7.485714	7.485714	9955400	7.6	
4	2010-01-08	7.498571	7.742857	7.465714	7.614286	7.614286	8180900	7.5	
...	...	...	...	...	...	...	...	...	
3114	2022-05-17	189.169998	191.399994	185.169998	190.559998	190.559998	9876700	186.5	
3115	2022-05-18	186.720001	187.699997	176.270004	177.190002	177.190002	9665600	189.7	
3116	2022-05-19	178.050003	186.300003	175.710007	183.479996	183.479996	10448500	184.6	
3117	2022-05-20	185.869995	190.190002	179.770004	186.350006	186.350006	10422600	183.5	
3118	2022-05-23	186.149994	187.660004	177.889999	187.440002	187.440002	9558000	183.4	

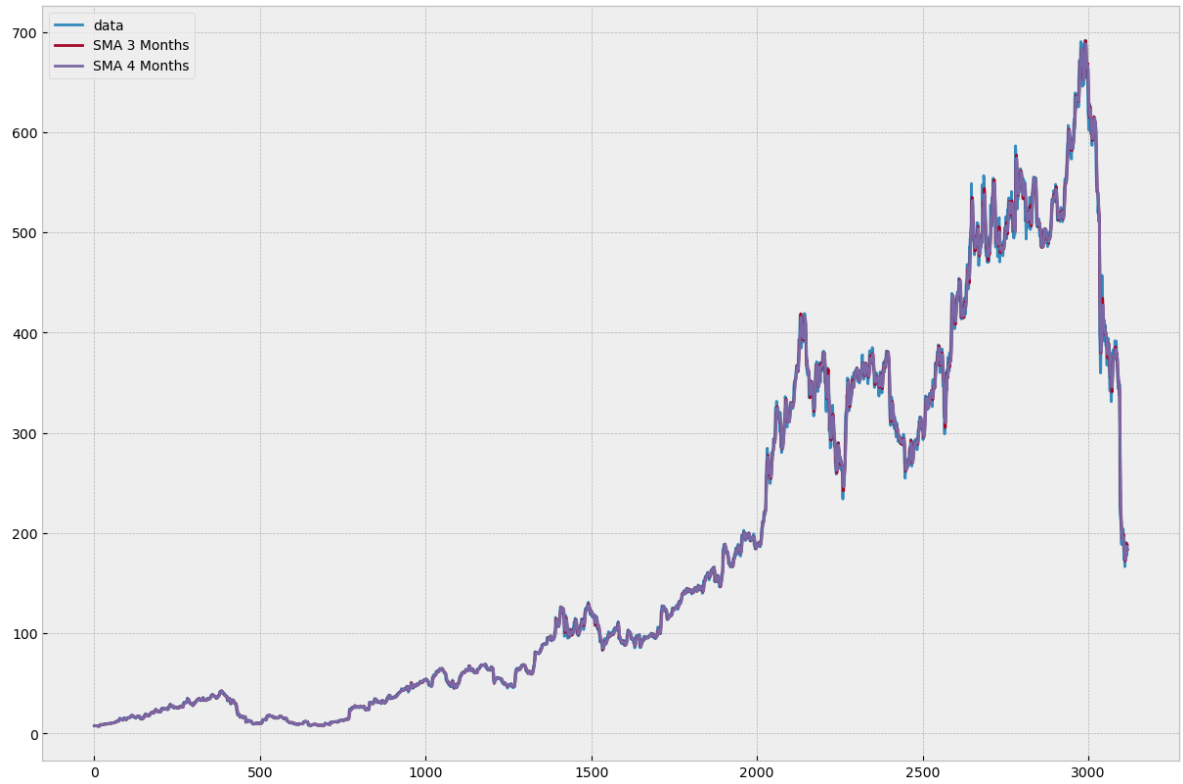
3119 rows × 9 columns

In [22]:

```
plt . figure ( figsize =[15 ,10])
plt . grid ( True )
plt . plot ( df ['Close'] , label ='data')
plt . plot ( df ['SMA_3'] , label ='SMA 3 Months')
plt . plot ( df ['SMA_4'] , label ='SMA 4 Months')
plt . legend ( loc =2)
```

Out[22]:

```
<matplotlib.legend.Legend at 0x1f6ac57f100>
```



In [ ]:

```
```