STOCK MARKET PREDICTION

Stock prediction of Tata Motors

In [1]:
```
#install the dependencies
```

In [6]:
```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

In [7]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [9]:
```
dataset = pd.read_csv('TTM.csv')
```

In [10]:
```
dataset.describe()
```

Out[10]:

	Open	High	Low	Close	Adj Close	Volume
count	253.000000	253.000000	253.000000	253.000000	253.000000	2.530000e+02
mean	12.534348	12.704783	12.360514	12.550318	12.550318	1.973424e+06
std	5.809727	5.871121	5.738625	5.823067	5.823067	1.110034e+06
min	5.390000	5.570000	5.100000	5.320000	5.320000	1.650800e+05
25%	7.740000	7.860000	7.680000	7.800000	7.800000	1.132100e+06
50%	10.000000	10.130000	9.840000	10.030000	10.030000	1.713300e+06
75%	19.350000	19.639999	19.129999	19.459999	19.459999	2.466500e+06
max	23.450001	23.700001	23.190001	23.420000	23.420000	7.612300e+06

In [12]:
```
x = dataset[['High','Low','Open','Volume']].values
y = dataset['Close'].values
```

In [13]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=0)
```

In [14]:
```
regressor = LinearRegression()
```

In [15]:
```
regressor.fit(x_train,y_train)
```

Out[15]:
```
LinearRegression()
```

In [16]:
```
print(regressor.coef_)
```

```
[ 8.49671854e-01  5.56820209e-01 -4.06071568e-01  1.80722215e-10]
```

In [18]:
```
print(regressor.intercept_)
```

```
-0.03483596200471162
```

In [19]:
```
predicted = regressor.predict(x_test)
```

In [21]:
```
dframe = pd.DataFrame({'Actual':y_test.flatten(),'Predicted':predicted.flatten()})
```

In [23]:
```
dframe.head(25)
```

Out[23]:

	Actual	Predicted
0	12.310000	12.466940
1	9.770000	9.781015
2	13.340000	13.374007
3	8.980000	9.005619
4	12.520000	12.346514
5	21.870001	21.816597
6	8.840000	8.879158
7	20.520000	20.483300

	Actual	Predicted
8	7.540000	7.501049
9	10.030000	9.968066
10	12.510000	12.506655
11	20.450001	20.491690
12	9.670000	9.603011
13	22.170000	21.988308
14	11.990000	11.949105
15	8.390000	8.405171
16	21.990000	22.001350
17	13.450000	13.406676
18	12.570000	12.502329
19	9.110000	9.054586
20	19.520000	19.474344
21	7.230000	7.208628
22	20.510000	20.618768
23	11.630000	11.823319
24	9.580000	9.506163

In [27]:
```
print('Mean Standard Error:',metrics.mean_absolute_error(y_test,predicted))
print('Mean Squared Error:',metrics.mean_squared_error(y_test,predicted))
print('Root Mean Squared Error:',math.sqrt(metrics.mean_squared_error(y_test,predicted)))
```

```
Mean Standard Error: 0.0811555114607198
Mean Squared Error: 0.01637738658035304
Root Mean Squared Error: 0.1279741637220304
```

In [25]:
```
import math
```

In [28]:
```
graph = dframe.head(20)
```

In [29]:
```
graph.plot(kind='bar')
```

Out[29]: <AxesSubplot:>