In [1]:

```
import pandas as pd
import numpy as np
from sklearn import metrics
%matplotlib inline
import matplotlib.pyplot as plt
```

In [2]:

```
dataset=pd.read_csv(r"C:\Users\SWAJAN\Documents\education\da project\datasets\nifty_it_index.csv")
```

In [3]:

```
dataset.head()
```

Out[3]:

| | Date | Open | High | Low | Close | Volume | Turnover |
|---|---|---|---|---|---|---|---|
| **0** | 2015-01-01 | 11214.80 | 11235.75 | 11166.35 | 11215.70 | 4246150 | 3.575100e+09 |
| **1** | 2015-01-02 | 11214.65 | 11399.10 | 11214.65 | 11372.10 | 10004862 | 9.645600e+09 |
| **2** | 2015-01-05 | 11369.35 | 11433.75 | 11186.95 | 11248.55 | 8858018 | 1.059000e+10 |
| **3** | 2015-01-06 | 11186.10 | 11186.10 | 10909.00 | 10959.90 | 12515739 | 1.364500e+10 |
| **4** | 2015-01-07 | 11013.20 | 11042.35 | 10889.55 | 10916.00 | 10976356 | 1.203440e+10 |

In [4]:

```
dataset.shape
```

Out[4]:

```
(248, 7)
```

In [5]:

```
dataset.isnull().sum()
```

Out[5]:

```
Date        0
Open        0
High        0
Low         0
Close       0
Volume      0
Turnover    0
dtype: int64
```

In [6]:

```
dataset.isna().any()
```

Out[6]:

```
Date        False
Open        False
High        False
Low         False
Close       False
Volume      False
Turnover    False
dtype: bool
```

In [7]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 248 entries, 0 to 247
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Date      248 non-null    object 
 1   Open      248 non-null    float64
 2   High      248 non-null    float64
 3   Low       248 non-null    float64
 4   Close     248 non-null    float64
 5   Volume    248 non-null    int64  
 6   Turnover  248 non-null    float64
dtypes: float64(5), int64(1), object(1)
memory usage: 13.7+ KB
```

In [8]:

```
dataset.describe()
```

Out[8]:

| | Open | High | Low | Close | Volume | Turnover |
|---|---|---|---|---|---|---|

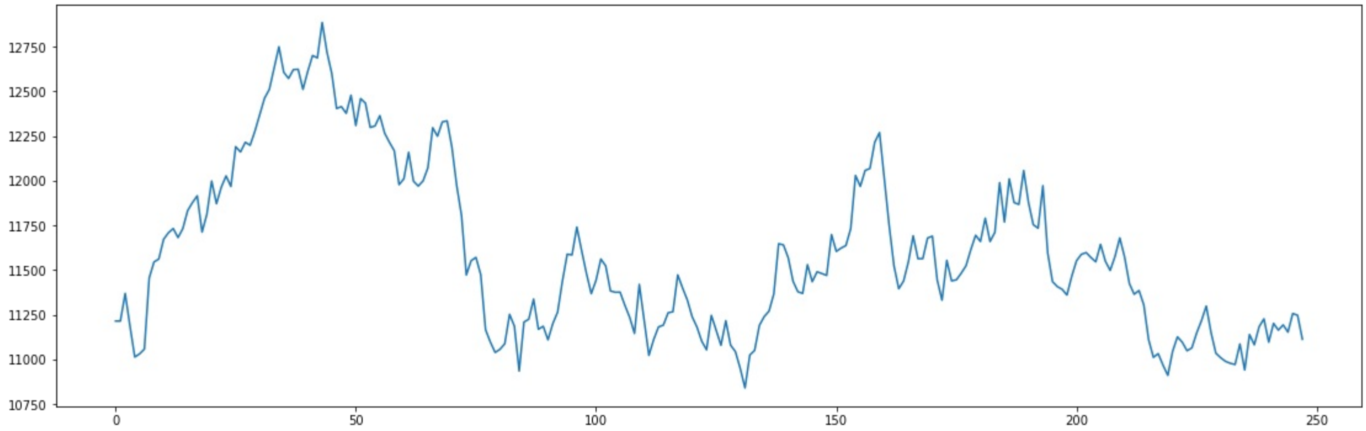
| | | | | | | |
|---|---|---|---|---|---|---|
| **count** | 248.000000 | 248.000000 | 248.000000 | 248.000000 | 2.480000e+02 | 2.480000e+02 |
| **mean** | 11601.495968 | 11673.756250 | 11505.632056 | 11585.626613 | 1.383053e+07 | 1.354940e+10 |
| **std** | 468.997883 | 472.763542 | 462.203401 | 466.678465 | 6.401886e+06 | 5.461539e+09 |
| **min** | 10840.650000 | 10950.250000 | 10759.850000 | 10798.250000 | 7.952400e+05 | 8.272000e+08 |
| **25%** | 11214.762500 | 11268.200000 | 11133.312500 | 11210.200000 | 9.304708e+06 | 9.438500e+09 |
| **50%** | 11524.625000 | 11578.075000 | 11418.975000 | 11503.850000 | 1.218344e+07 | 1.259385e+10 |
| **75%** | 11927.637500 | 11999.187500 | 11787.050000 | 11886.337500 | 1.667710e+07 | 1.657345e+10 |
| **max** | 12885.750000 | 12908.100000 | 12635.500000 | 12855.900000 | 4.461970e+07 | 3.685160e+10 |

In [9]:

```
dataset['Open'].plot(figsize=(19,6))
```

Out[9]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1864ac14040>
```



In [12]:

```
import seaborn as sns
```

In [10]:

```
x=dataset[['Open','High','Low','Volume']]
y=dataset['Close']
```

In [11]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train , y_test= train_test_split(x,y,test_size=0.1,random_state=0)
```

In [12]:

```
x_train.shape
x_test.shape
```

Out[12]:

```
(25, 4)
```

In [14]:

```
x_train.shape
```

Out[14]:

```
(223, 4)
```

In [15]:

```
from sklearn.linear_model import LinearRegression
```

In [16]:

```
 from sklearn.metrics import accuracy_score
```

In [17]:

```
reg=LinearRegression()
```

In [18]:

```
reg.fit(x_train,y_train)
```

Out[18]:

```
LinearRegression()
```

In [19]:

```
reg.coef_
```

Out[19]:
```
array([-6.12416370e-01,  6.86425217e-01,  9.14675534e-01,  1.78788655e-06])
```

In [20]:
```
reg.intercept_
```

Out[20]:
```
129.49177111783683
```

In [21]:
```
predicted=reg.predict(x_test)
```

In [22]:
```
predicted.shape
```

Out[22]:
```
(25,)
```

In [23]:
```
dframe=pd.DataFrame( y_test, predicted )
```

In [24]:
```
dfr=pd.DataFrame( { 'Actual Price':y_test, 'Predicted price': predicted } )
```

In [25]:
```
dfr.head()
```

Out[25]:

| | Actual Price | Predicted price |
|---|---|---|
| **247** | 11212.55 | 11214.351164 |
| **168** | 11604.70 | 11668.027718 |
| **76** | 11132.25 | 11203.643033 |
| **150** | 11597.05 | 11633.932097 |
| **145** | 11594.15 | 11582.369420 |

In [26]:
```
reg.score(x_test,y_test)
```

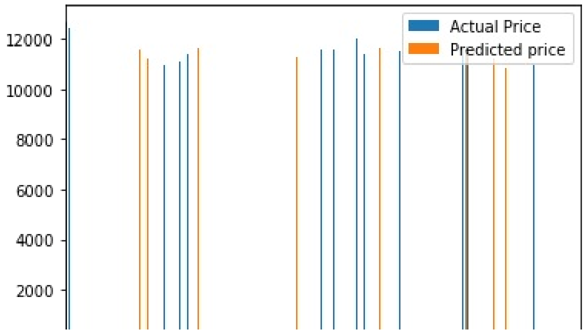
Out[26]:
```
0.9910198708660388
```

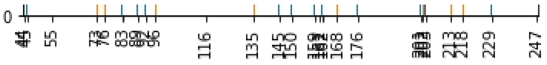
In [27]:
```
import math

print('mean absolute 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 absolute error :  32.184875678432654
mean squared error :  1670.777632579646
root mean squared error :  40.8751468814443
```

In [28]:
```
graph=dfr.head(50)
graph.plot(kind="bar")
```

Out[28]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1864ae443d0>
```



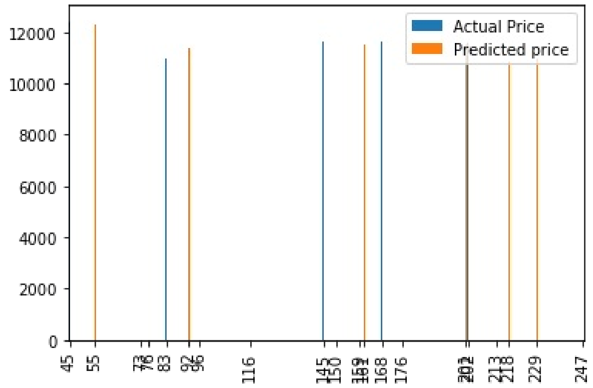


In [29]:

```
graph=dfr.head(20)
graph.plot(kind="bar")
```

Out[29]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1864af35e80>
```



In [30]:

```
dfr.head(50)
```

Out[30]:

| | Actual Price | Predicted price |
|---|---|---|
| **247** | 11212.55 | 11214.351164 |
| **168** | 11604.70 | 11668.027718 |
| **76** | 11132.25 | 11203.643033 |
| **150** | 11597.05 | 11633.932097 |
| **145** | 11594.15 | 11582.369420 |
| **73** | 11558.40 | 11560.279858 |
| **45** | 12422.25 | 12405.718197 |
| **159** | 11985.25 | 12006.174718 |
| **218** | 10836.70 | 10838.162612 |
| **213** | 11214.20 | 11223.022869 |
| **96** | 11623.35 | 11662.316897 |
| **201** | 11552.00 | 11517.979318 |
| **83** | 10938.85 | 10989.219872 |
| **176** | 11520.30 | 11464.017412 |
| **161** | 11455.15 | 11485.688927 |
| **202** | 11576.35 | 11563.274414 |
| **55** | 12244.45 | 12301.788924 |
| **116** | 11429.80 | 11401.642199 |
| **229** | 10969.05 | 10973.234465 |
| **92** | 11372.20 | 11405.774807 |
| **203** | 11583.55 | 11573.809102 |
| **135** | 11269.20 | 11295.440878 |
| **162** | 11376.75 | 11268.687372 |
| **89** | 11093.60 | 11059.133054 |
| **44** | 12665.50 | 12710.293541 |

In [31]:

```
reg.score(x_test,y_test)
reg.score(x_train, y_train)
```

Out[31]:

```
0.9938496329558479
```

In [32]:

```
from sklearn.metrics import r2 score
```

```
r2_score(y_test,predicted)
```

Out[32]:

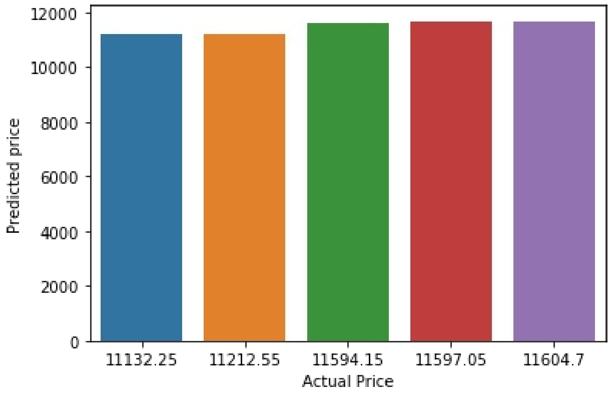
```
0.9910198708660388
```

In [34]:

```
import seaborn as sns
sns.barplot(x="Actual Price",y="Predicted price",data=dfr.head(5))
```

Out[34]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1864d714970>
```



In [ ]:

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