In [2]:
```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

from sklearn.neural_network import MLPRegressor

from sklearn import metrics

from sklearn.model_selection import GridSearchCV
```

In [4]:
```
df = pd.read_csv('C:\\Users\\kgnan\\Downloads\\house_prices.csv')
df = df.drop(['id', 'date', 'yr_built', 'yr_renovated', 'zipcode', 'lat', '
df.head()
```

Out[4]:

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition
0	221900.0	3	1.00	1180	5650	1.0	0	0	3
1	538000.0	3	2.25	2570	7242	2.0	0	0	3
2	180000.0	2	1.00	770	10000	1.0	0	0	3
3	604000.0	4	3.00	1960	5000	1.0	0	0	5
4	510000.0	3	2.00	1680	8080	1.0	0	0	3

In [10]:
```
#DATA PREPROCESSING
x = df.drop('price', axis=1)
y = df['price']

trainX, testX, trainY, testY = train_test_split(x, y, test_size = 0.2)
```

In [11]:
```
sc=StandardScaler()

scaler = sc.fit(trainX)
trainX_scaled = scaler.transform(trainX)
testX_scaled = scaler.transform(testX)
```

In [12]:
```
###MLPRegressor
mlp_reg = MLPRegressor(hidden_layer_sizes=(150,100,50),
                       max_iter = 300,activation = 'relu',
                       solver = 'adam')

mlp_reg.fit(trainX_scaled, trainY)
```

```
C:\Users\kgnan\anaconda3\lib\site-packages\sklearn\neural_network\_multila
yer_perceptron.py:582: ConvergenceWarning: Stochastic Optimizer: Maximum i
terations (300) reached and the optimization hasn't converged yet.
  warnings.warn(
```

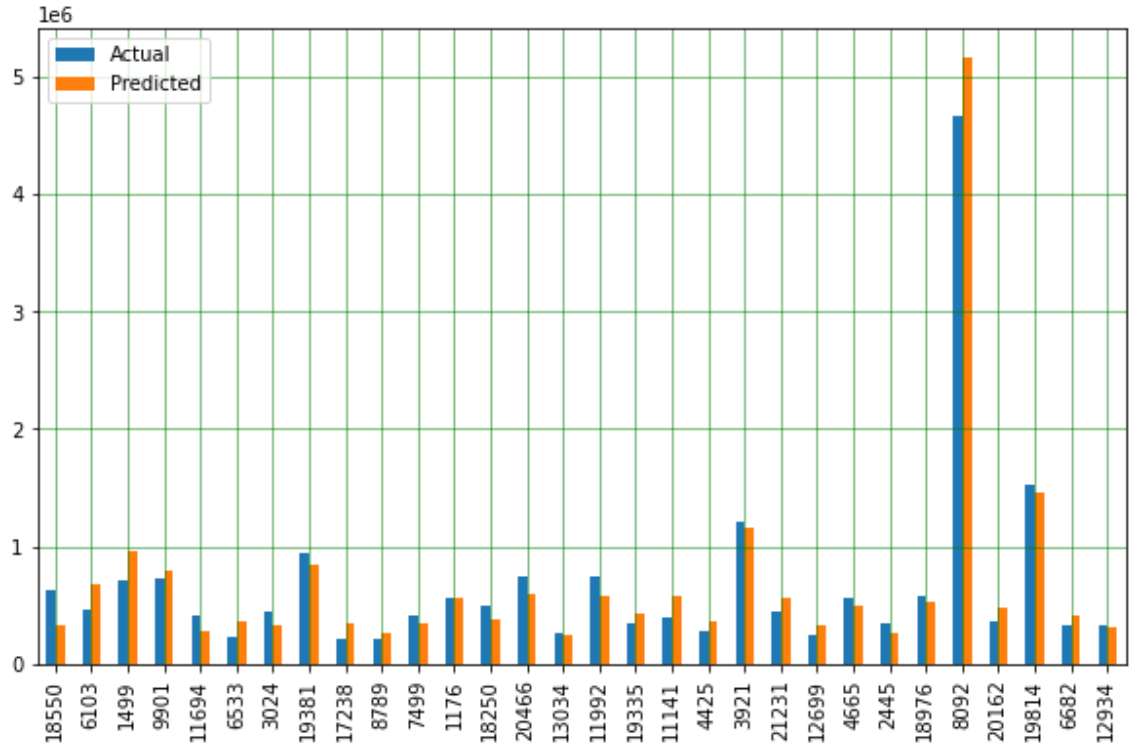
Out[12]:
```
MLPRegressor(hidden_layer_sizes=(150, 100, 50), max_iter=300)
```

In [16]:
```
###Model Evaluation
y_pred = mlp_reg.predict(testX_scaled)
df_temp = pd.DataFrame({'Actual': testY, 'Predicted': y_pred})
df_temp.head()
```

Out[16]:

	Actual	Predicted
18550	630000.0	331230.635449
6103	464950.0	677779.492193
1499	720000.0	964300.322978
9901	735000.0	787401.733963
11694	410000.0	287273.406049

In [17]:
```
df_temp = df_temp.head(30)
df_temp.plot(kind='bar',figsize=(10,6))
plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
plt.show()
```



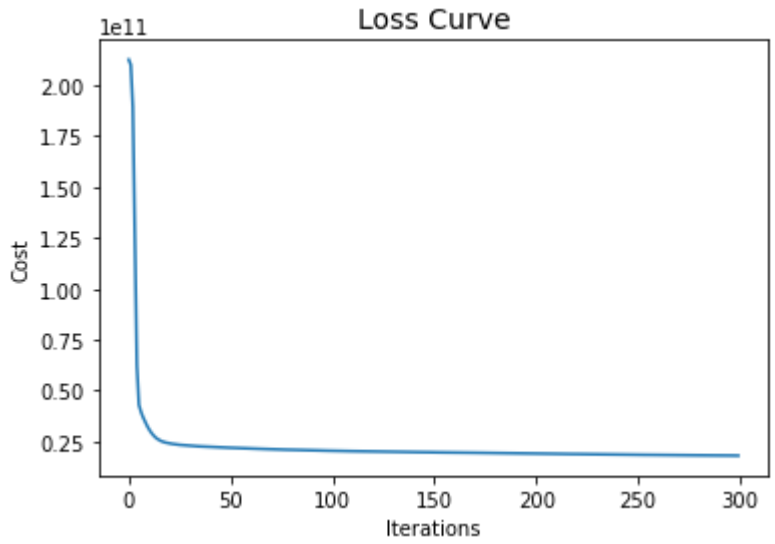
In [ ]:

In [ ]:

In [18]:
```
print('Mean Absolute Error:', metrics.mean_absolute_error(testY, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(testY, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(testY,
```

```
Mean Absolute Error: 127075.8528942391
Mean Squared Error: 37713355952.00089
Root Mean Squared Error: 194199.26867009795
```

In [19]:
```
plt.plot(mlp_reg.loss_curve_)
plt.title("Loss Curve", fontsize=14)
plt.xlabel('Iterations')
plt.ylabel('Cost')
plt.show()
```



In [ ]: