## # IMPORTING LIBRARIES

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import confusion\_matrix,accuracy\_score

import math

from sklearn import metrics

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestRegressor

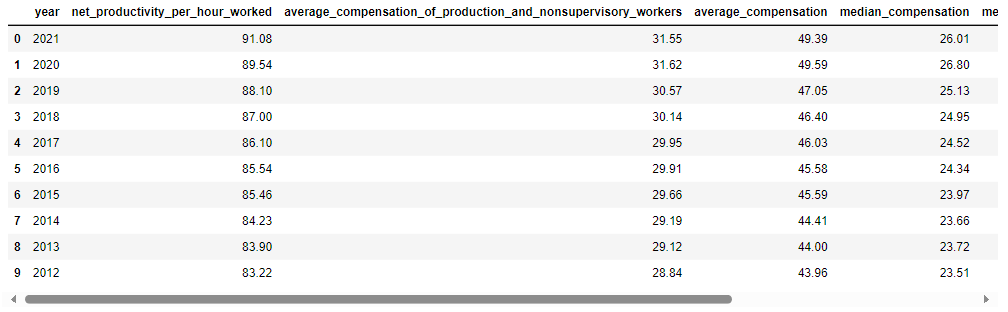
from sklearn.metrics import confusion\_matrix,accuracy\_score

# # READING DATASET

df=pd.read\_csv('productivity\_n\_hourly\_compensation.csv')

df

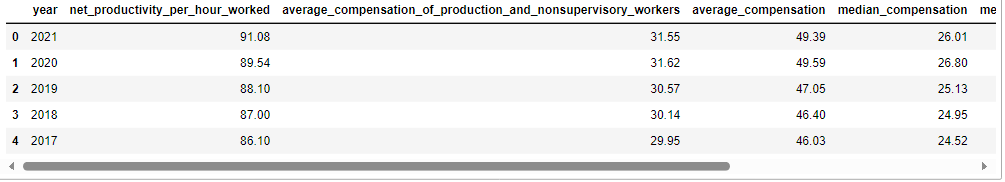
## output:



# # TO CHECK THE FIRST FIVE COLUMNS

df.head()

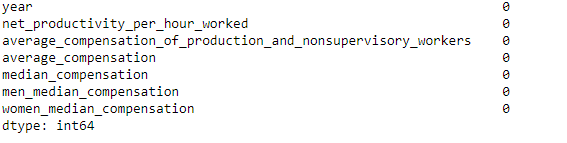
# output:



# # CHECKING MISSING VALUES

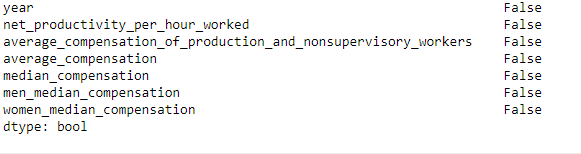
df.isnull().sum()

# output:



# df.isnull( ).any( )

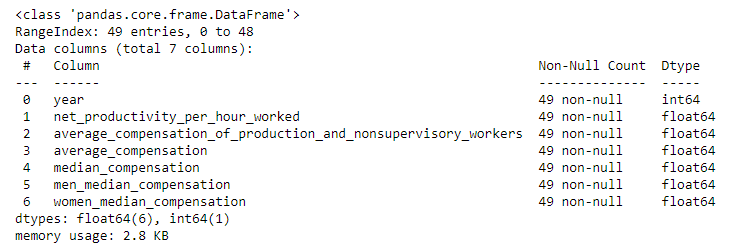
# output:



# # CHECKING INFORMATION ABOUT DATASET

df.info()

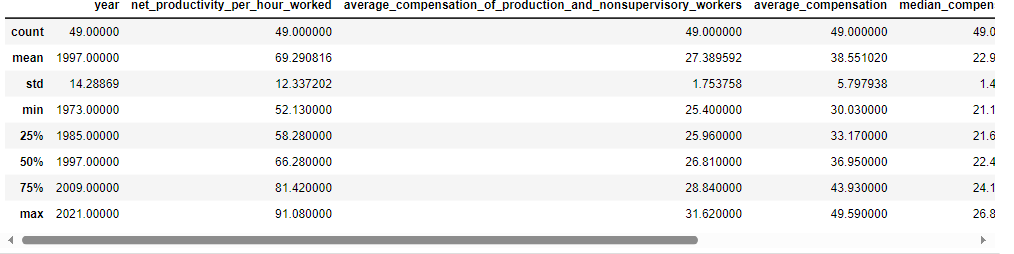
# output:



# #TO CHECK STATISTICS OF DATASET

df.describe()

# output:



print('lenghth of dataset:',len(df))

print('shape of dataset:',df.shape)

# output:



#

# PLOTING LINE GRAPH FOR AVERAGE COMPENSATION OF PRODUCTIVITY\_N\_HOURLY\_COMPENSATION

import matplotlib.pyplot as plt

df['average\_compensation'].plot(figsize=(16,6))

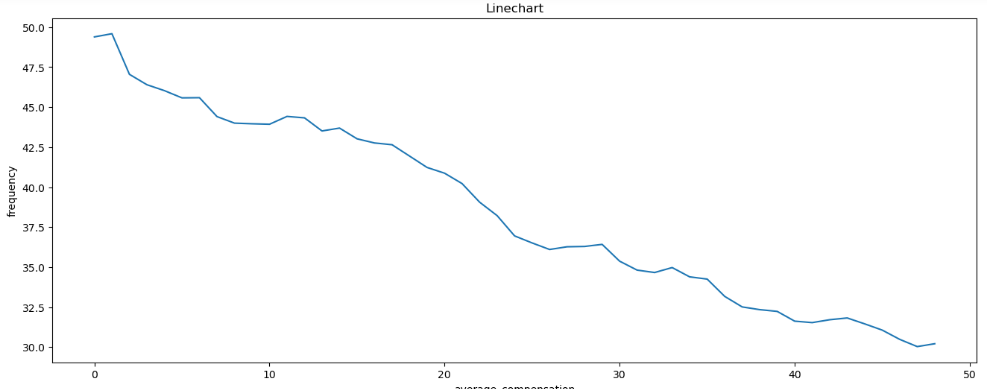
plt.title('Linechart')

plt.xlabel('average\_compensation')

plt.ylabel('frequency')

plt.show()

# output:



# # TRAINING THE MODEL FOR MultipleLinearRegression

# # splitting the dataset

x=df.drop('net\_productivity\_per\_hour\_worked',axis=1)

y=df['net\_productivity\_per\_hour\_worked']

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,random\_state=0)

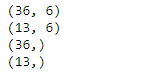
print(x\_train.shape)

print(x\_test.shape)

print(y\_train.shape)

print(y\_test.shape)

# output:

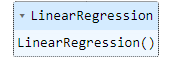


# # MODEL TRAINING

regressor=LinearRegression()

regressor.fit(x\_train,y\_train)

# output:



# #TO CHECK CO-EFFICIENT AND INTERCEPT

print(regressor.coef\_)

print(regressor.intercept\_)

# output:



# # PREDICTION

predicted=regressor.predict(x\_test)

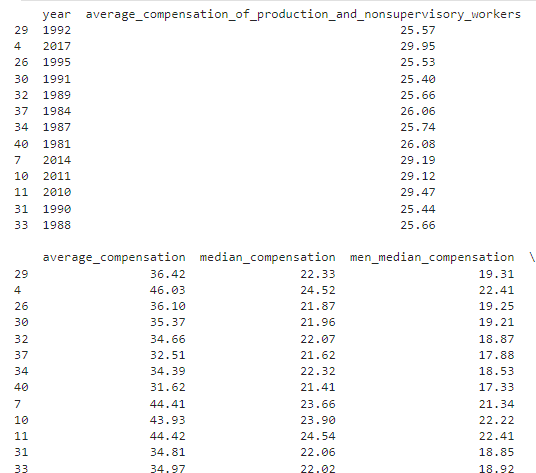
predicted

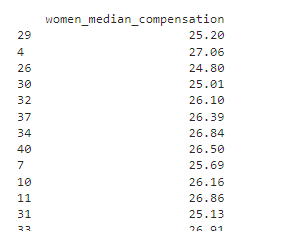
# output:



print(x\_test)

# output:





predicted.shape

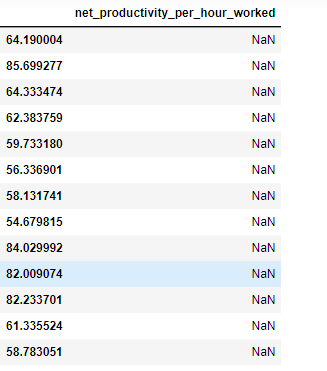
# output:

(13,)

dframe=pd.DataFrame(y\_test,predicted)

dframe

# output:

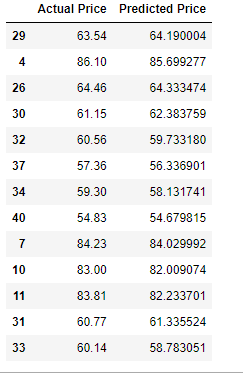


# # TO CHECK ACTUAL PRICE AND PREDICTED PRICE

dfr=pd.DataFrame({'Actual Price':y\_test,'Predicted Price':predicted})

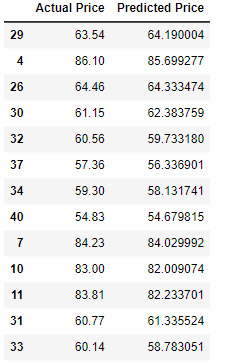
dfr

# output:



dfr.head(40)

# output:



# # EVALUATING THE MODEL

from sklearn.metrics import confusion\_matrix,accuracy\_score

train\_accuracy=regressor.score(x\_train,y\_train)

print('train\_accuracy:',train\_accuracy)

test\_accuracy=regressor.score(x\_test,y\_test)

print('test\_accuracy:',test\_accuracy)

# output:



import math

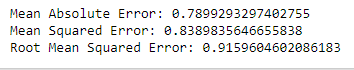
from sklearn import metrics

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)))

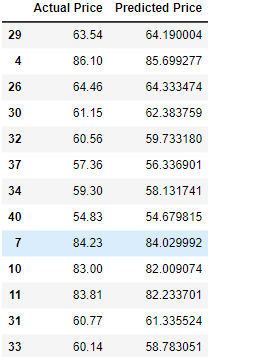
# output:



graph=dfr.head(20)

graph

# output:



# # TO PLOT THE BAR GRAPH AND TO CHECK THE ACTUAL PRICE AND PREDICTED PRICE

graph.plot(kind='bar')

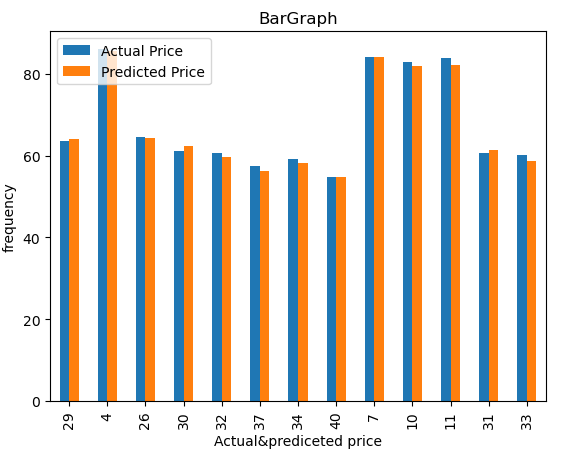
plt.title('BarGraph')

plt.xlabel('Actual&prediceted price')

plt.ylabel('frequency')

plt.show()

# output:



**from** sklearn.model\_selection **import** train\_test\_split

x\_train,x\_test,y\_train,y\_test**=**train\_test\_split(x,y,test\_size**=**0.90,random\_state**=**100)

print(x\_train**.**shape)

print(x\_test**.**shape)

print(y\_train**.**shape)

print(y\_test**.**shape)

# output:



# # TRAINING THE MODEL FOR RandomForestRegressor

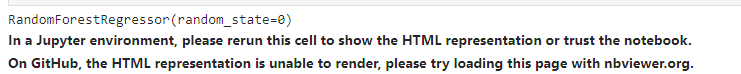
# #MODEL TRAINING

**from** sklearn.ensemble **import** RandomForestRegressor

regressorRandomForestRegressor(n\_estimators**=**100,random\_state**=**0)

regressor**.**fit(x\_train,y\_train)

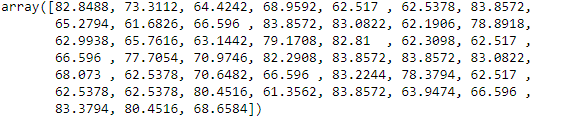
# output:



predicted=regressor.predict(x\_test)

predicted

# output:



from sklearn.metrics import confusion\_matrix,accuracy\_score

train\_accuracy=regressor.score(x\_train,y\_train)

print('train\_accuracy:',train\_accuracy)

R\_test\_accuracy=regressor.score(x\_test,y\_test)

print('test\_accuracy:',R\_test\_accuracy)

# output:



# # USING BAR GRAPH COMPARISON BETWEEN LINEAR MODEL AND RANDOM FOREST REGRESSION

import matplotlib.pyplot as plt

linear\_regression\_accuracy = 0.9949918060732188

random\_forest\_accuracy =0.8404782538548659

accuracy\_scores = [linear\_regression\_accuracy, random\_forest\_accuracy]

model\_names = ['Linear Regression', 'Random Forest Regression']

plt.bar(model\_names, accuracy\_scores)

plt.xlabel('Regression Models')

plt.ylabel('Test Accuracy')

plt.title('Comparison of Test Accuracy: Linear Regression vs Random Forest Regression')

plt.show()

# output:

