# productivity\_n\_hourly\_compensation

# ABSTRACTION:

Productivity and hourly compensation are key economic indicators that are often used to assess the efficiency and income levels in an economy. Here's a brief explanation of each term:

1. Productivity:

Productivity refers to the efficiency with which resources (such as labor, capital, and technology) are utilized to produce goods and services. It measures the output of goods and services relative to the input of resources. Higher productivity indicates that an economy or a company is producing more with the same amount of resources or the same with fewer resources, which is generally seen as a positive sign. Productivity can be measured in various ways, but one common measure is labor productivity, which compares the output of goods and services to the amount of labor input (usually measured in hours or workers). Higher labor productivity means that workers are producing more output per hour worked.

2. Hourly Compensation:

Hourly compensation, also known as hourly wage or earnings, is the amount of money paid to workers for each hour of work. It includes all forms of compensation, such as base wages, overtime pay, bonuses, and benefits. Hourly compensation can vary significantly across different industries, job roles, and regions.Hourly compensation is an important metric for assessing the income earned by workers. It is often used to gauge the standard of living and economic well-being of a country's workforce. Trends in hourly compensation can indicate whether workers are experiencing wage growth, stagnation, or decline over time.

Information about productivity and hourly compensation can be obtained from various sources, including government agencies, labor market reports, and economic research organizations. Some commonly used sources for this data in the United States, for example, include the Bureau of Labor Statistics (BLS) and the Bureau of Economic Analysis (BEA). These agencies provide regular reports and statistics on labor productivity, wage levels, and other economic indicators.

# METHODOLOGY:

# Multiple linear regression:

Multiple linear regression (MLR), also known simply as multiple

regression, is a **statistical technique that uses several explanatory variables to predict the outcome of a response variable**. Multiple regression is an extension of linear (OLS) regression that uses just one explanatory variable.

# RandomForestRegression:

RandomForestRegression is a supervised learning algorithm and bagging technique that uses an ensemble learning method for regression in machine learning.

# ANN model:

**An Artificial Neural Network (ANN) is a computational model inspired by the structure and functioning of the human brain. It is a type of machine learning algorithm that is designed to recognize patterns and make predictions or decisions based on input data. ANNs can be used for various tasks, including classification, regression, and pattern recognition. They have been successfully applied in areas such as image and speech recognition, natural language processing, financial forecasting, and many other domains.**

# # 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

from sklearn.metrics import mean\_squared\_error,mean\_absolute\_error

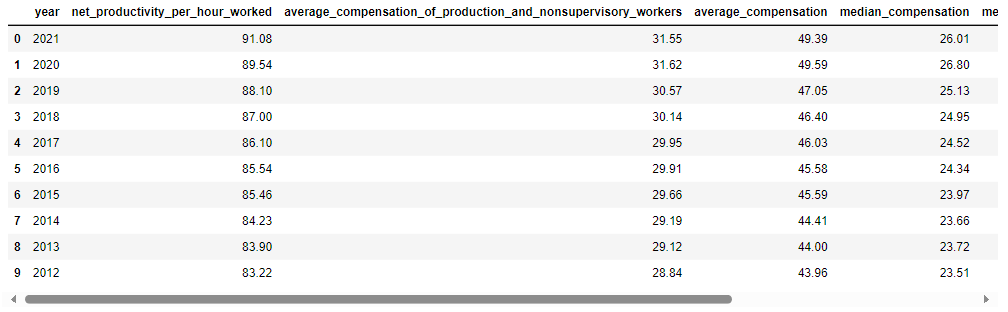
from sklearn.ensemble import RandomForestRegressorv

# # READING DATASET

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

df

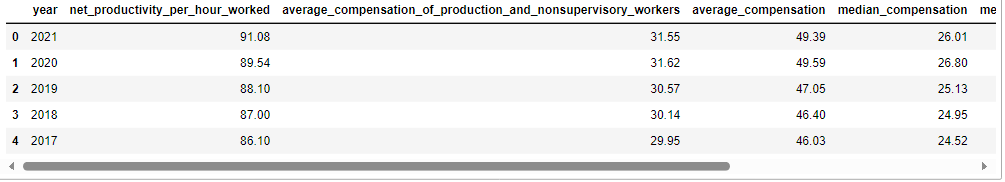
## output:



# # TO CHECK THE FIRST FIVE COLUMNS

df.head()

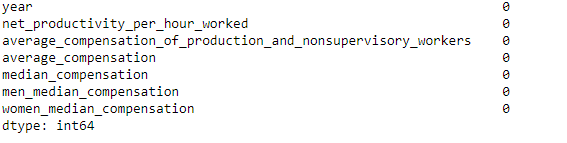
# output:



# # CHECKING MISSING VALUES

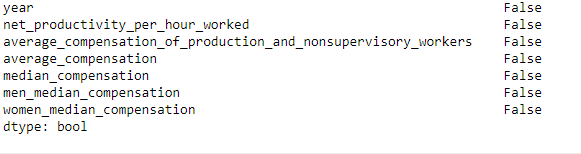
df.isnull().sum()

# output:



# df.isnull( ).any( )

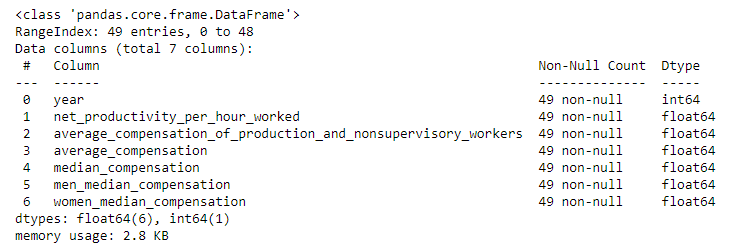
# output:



# # CHECKING INFORMATION ABOUT DATASET

df.info()

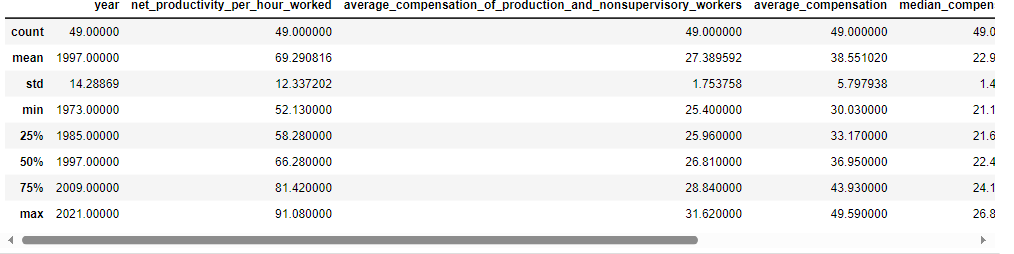
# output:



# #TO CHECK STATISTICS OF DATASET

df.describe()

# output:



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

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

# output:



# 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)

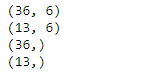
print(x\_train.shape)

print(x\_test.shape)

print(y\_train.shape)

print(y\_test.shape)

# output:

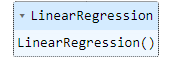


# # MODEL TRAINING

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)

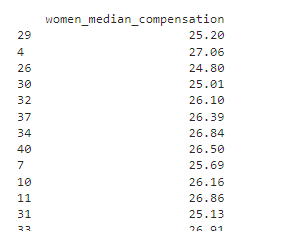
predicted

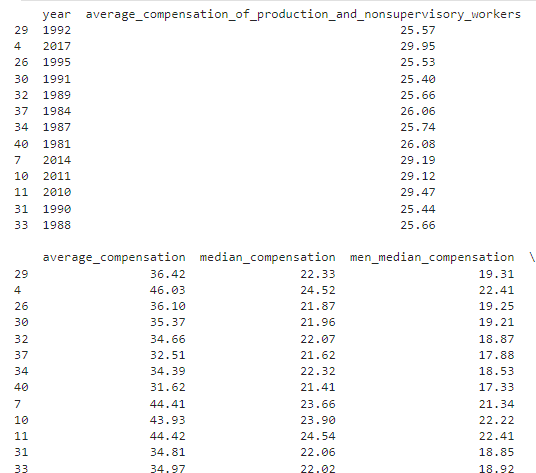
# output:



print(x\_test)

# output:





predicted.shape

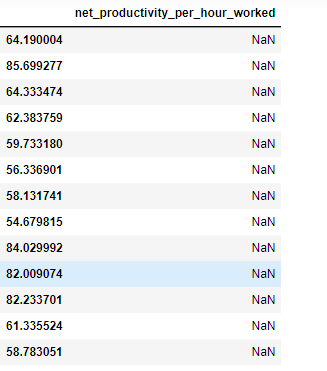
# output:

(13,)

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

dframe

# output:

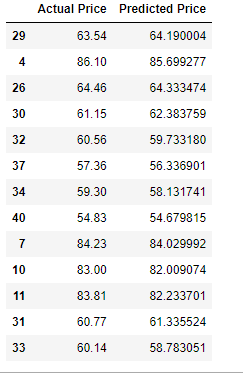


# # TO CHECK ACTUAL PRICE AND PREDICTED PRICE

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

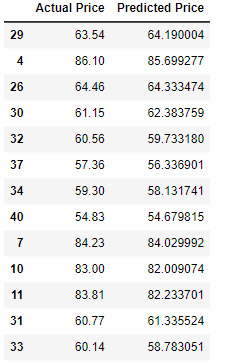
dfr

# 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

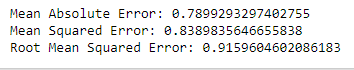
from sklearn import metrics

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)))

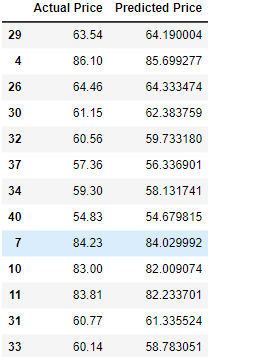
# output:



graph=dfr.head(20)

graph

# output:



**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)

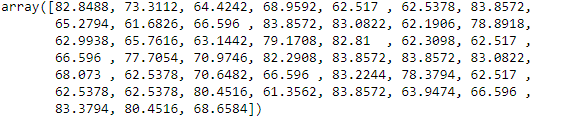
# output:



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:



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

graph.plot(kind='bar')

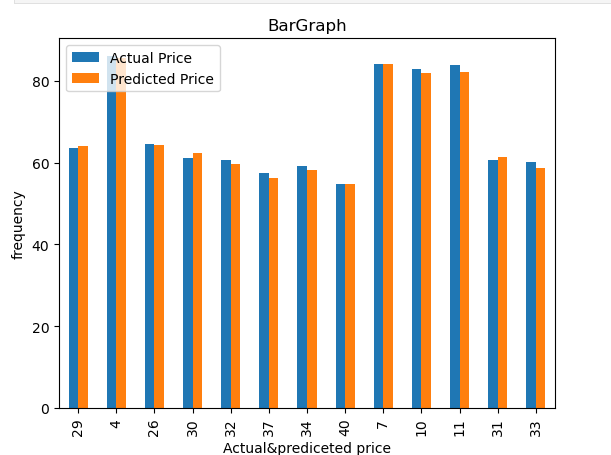
plt.title('BarGraph')

plt.xlabel('Actual&prediceted price')

plt.ylabel('frequency')

plt.show()

# output:



# Result:

**In the above graph x-axis represents the actual and** prediceted price and y-axis represent the frequency. It comparision between **the actual and** prediceted price,in this graph prediction price is almost same as actual price

# # 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:



# Prediction

predicted=regressor.predict(x\_test)

predicted

# evaluating the model

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

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

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

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

# output:



# Preprocessing the x\_train and x\_test

scaler=StandardScaler()

x\_train=scaler.fit\_transform(x\_train)

x\_test=scaler.transform(x\_test)

# ­­­ ANN Model

# training the model using ANN model

ann\_model=MLPRegressor(hidden\_layer\_sizes=(128,64,32),activation='relu',solver='lbfgs')

ann\_model.fit(x\_train,y\_train)

# output:

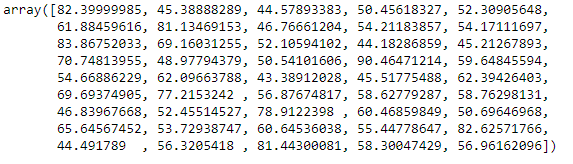


# Prediction

predicted=ann\_model.predict(x\_test)

predicted

# output:



# evaluating the model

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

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

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

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

print('Mean Absolute Error:',metrics.mean\_absolute\_error(y\_test,y\_pred))

print('Mean Squared Error:',metrics.mean\_squared\_error(y\_test,y\_pred))

print('Root Mean Squared Error:',math.sqrt(metrics.mean\_squared\_error(y\_test,y\_pred)))

# output:

train\_accuracy(R\_Squered): 0.9999999999997117

test\_accuracy(R\_Squered): 0.9988898217061862

In [40]:



COMPARISON BETWEEN LINEAR MODEL , RANDOM FOREST REGRESSION and ANN Model using Barplot

import matplotlib.pyplot as plt

linear\_regression\_accuracy =0.9934898217061862

Random\_forest\_accuracy =0.8404782538548659

ANN= 0.998898217061862

accuracy\_scores = [linear\_regression\_accuracy, Random\_forest\_accuracy,ANN]

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

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

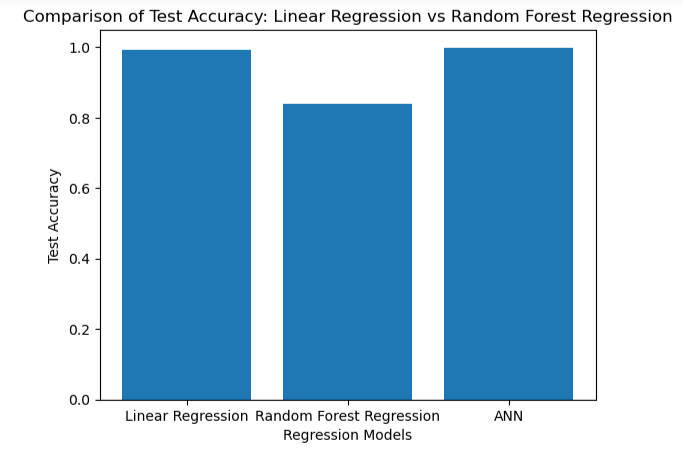
plt.xlabel('Regression Models')

plt.ylabel('Test Accuracy')

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

plt.show()

# output:



Result:

In this case, the x-axis might represent the different models (linear regression, Random forest regression and ANN model), while the y-axis represents a performance metric or evaluation criterion. We have compared test accuracy(R-squared) between Linear Regression model, Random Forest Regression model and ANN model using pictorial representation is shown below.

linear\_regression\_accuracy =0.9934898217061862

Random\_forest\_accuracy =0.8404782538548659

ANN Model= 0.998898217061862

After compering the test accuracy between these models,Ann Model has given the best accuracy.

Conclusion:

In this model we can use import the libraries and read the dataset,check the missing values. In the stage of training the model, it has been splitted the train and test dataset,and we also applied a linear regression algorithm to model to tarin it. Also we applied the random forest regressor algorithm and find the accuracy .Then we compared linear regression and random forest regressor using graph,in this comparision we got the best accuracy in linear regression.In the stage of training the model , we have splited the dataset into train and test dataset, and we applied LinearRegression , RandomForest regressor and ANN model and found train and test accuracy for all three models. Then we compared linear regression, random forest regressor and ANN model using bar graph.

In this comparision, we got the best accuracy in ANN Mode