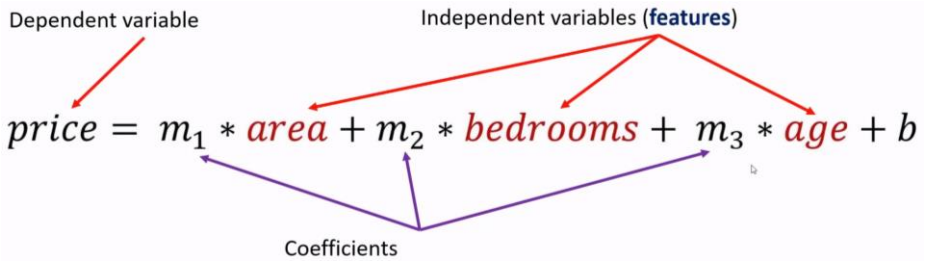
Home prices in Monroe Township, NJ (USA)

area	bedrooms	age	price
2600	3	20	550000
3000	4	15	565000
3200		18	610000
3600	3	30	595000
4000	5	8	760000

Given these home prices find out price of a home that has,

3000 sqr ft area, 3 bedrooms, 40 year old
2500 sqr ft area, 4 bedrooms, 5 year old

$$price = m_1 * area + m_2 * bedrooms + m_3 * age + b$$



$$y = m_1 x_1 + m_2 x_2 + m_3 x_3 + b$$

```
import pandas as pd
import numpy as np
from sklearn import linear_model
```

```
df = pd.read_csv("homeprices.csv")
df
```

	area	bedrooms	age	price
0	2600	3.0	20	550000
1	3000	4.0	15	565000
2	3200	NaN	18	610000
3	3600	3.0	30	595000
4	4000	5.0	8	760000

```
import math
median_bedrooms = math.floor(df.bedrooms.median())
median_bedrooms
```

```
3
```

```
df.bedrooms = df.bedrooms.fillna(median_bedrooms)
df
```

	area	bedrooms	age	price
0	2600	3.0	20	550000
1	3000	4.0	15	565000
2	3200	3.0	18	610000
3	3600	3.0	30	595000
4	4000	5.0	8	760000

```
reg = linear_model.LinearRegression()
reg.fit(df[['area','bedrooms','age']],df.price)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

```
reg.coef_
```

```
array([   137.25, -26025.  ,  -6825.  ])
```

```
reg.intercept_
```

```
383724.99999999983
```

```
reg.predict([[3000,3,40]])
```

```
array([ 444400.])
```

```
137.25*3000+-26025*3+-6825*40+383724.99999999983
```

```
444399.9999999998
```

```
reg.predict([[2500,4,5]])
```

```
array([ 588625.])
```

Combined Cycle Power Plant Data Set

Download: Data Folder, Data Set Description

Abstract: The dataset contains 9568 data points collected from a Combined Cycle Power Plant over 6 years (2006-2011), when the plant was set to work with full load.

Data Set Characteristics:	Multivariate	**Number of Instances:**	9568	**Area:**	Computer
Attribute Characteristics:	Real	**Number of Attributes:**	4	**Date Donated**	2014-03-26
Associated Tasks:	Regression	**Missing Values?**	N/A	**Number of Web Hits:**	185312

Source:

Pınar Tüfekci, Çorlu Faculty of Engineering, Namık Kemal University, TR-59860 Çorlu, Tekirdağ, Turkey
Email: ptufekci '@' nku.edu.tr

Heysem Kaya, Department of Computer Engineering, Boğaziçi University, TR-34342, Beşiktaş, İstanbul, Turkey
Email: heysem '@' boun.edu.tr

Data Set Information:

The dataset contains 9568 data points collected from a Combined Cycle Power Plant over 6 years (2006-2011), when the power plant was set to work with full load. Features consist of hourly average ambient variables Temperature (T), Ambient Pressure (AP), Relative Humidity (RH) and Exhaust Vacuum (V) to predict the net hourly electrical energy output (EP) of the plant.
A combined cycle power plant (CCPP) is composed of gas turbines (GT), steam turbines (ST) and heat recovery steam generators. In a CCPP, the electricity is generated by gas and steam turbines, which are combined in one cycle, and is transferred from one turbine to another. While the Vacuum is colected from and has effect on the Steam Turbine, he other three of the ambient variables effect the GT performance.
For comparability with our baseline studies, and to allow 5x2 fold statistical tests be carried out, we provide the data shuffled five times. For each shuffling 2-fold CV is carried out and the resulting 10 measurements are used for statistical testing.
We provide the data both in .ods and in .xlsx formats.

The dataset contains 9568 data points collected from a Combined Cycle Power Plant over 6 years (2006-2011), when the power plant was set to work with full load. Features consist of hourly average ambient variables Temperature (T), Ambient Pressure (AP), Relative Humidity (RH) and Exhaust Vacuum (V) to predict the net hourly electrical energy output (EP) of the plant.
A combined cycle power plant (CCPP) is composed of gas turbines (GT), steam turbines (ST) and heat recovery steam generators. In a CCPP, the electricity is generated by gas and steam turbines, which are combined in one cycle, and is transferred from one turbine to another. While the Vacuum is colected from and has effect on the Steam Turbine, he other three of the ambient variables effect the GT performance.
For comparability with our baseline studies, and to allow 5x2 fold statistical tests be carried out, we provide the data shuffled five times. For each shuffling 2-fold CV is carried out and the resulting 10 measurements are used for statistical testing.
We provide the data both in .ods and in .xlsx formats.

Attribute Information:

Features consist of hourly average ambient variables
- Temperature (T) in the range 1.81°C and 37.11°C,
- Ambient Pressure (AP) in the range 992.89-1033.30 milibar,
- Relative Humidity (RH) in the range 25.56% to 100.16%
- Exhaust Vacuum (V) in teh range 25.36-81.56 cm Hg
- Net hourly electrical energy output (EP) 420.26-495.76 MW

The averages are taken from various sensors located around the plant that record the ambient variables every second. The variables are given without normalization.

Import Libraries

Import dataset

Define x and y

Split the dataset in training set and test set

Train the model on the training set

Predict the test set results

Evaluate the model

Plot the results

Predicted values

Import Libraries

```
import pandas as pd
import numpy as np
```

Import dataset

```
data_df=pd.read_csv('/Users/megha/MLRPractice.csv')
```

```
data_df.head()
```

	AT	V	AP	RH	PE
0	14.96	41.76	1024.07	73.17	463.26
1	25.18	62.96	1020.04	59.08	444.37
2	5.11	39.40	1012.16	92.14	488.56
3	20.86	57.32	1010.24	76.64	446.48
4	10.82	37.50	1009.23	96.62	473.90

Define x and y

```
x=data_df.drop(['PE'],axis=1).values
y=data_df['PE'].values
```

```
print(x)
```

```
[[  14.96   41.76 1024.07   73.17]
 [  25.18   62.96 1020.04   59.08]
 [   5.11   39.4  1012.16   92.14]
 ...
 [  31.32   74.33 1012.92   36.48]
 [  24.48   69.45 1013.86   62.39]
 [  21.6    62.52 1017.23   67.87]]
```

```
print(y)
```

```
[463.26 444.37 488.56 ... 429.57 435.74 453.28]
```

Split the dataset in training set and test set

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

Train the model on the training set

```
from sklearn.linear_model import LinearRegression
ml=LinearRegression()
ml.fit(x_train,y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Predict the test set results

```
y_pred=ml.predict(x_test)
print(y_pred)
```

```
[431.40245096 458.61474119 462.81967423 ... 432.47380825 436.16417243
 439.00714594]
```

Predict the test set results

```
y_pred=ml.predict(x_test)
print(y_pred)
```

```
[431.40245096 458.61474119 462.81967423 ... 432.47380825 436.16417243
 439.00714594]
```

```
ml.predict([[14.96,41.76,1024.07,73.17]])
```

```
array([467.34820092])
```

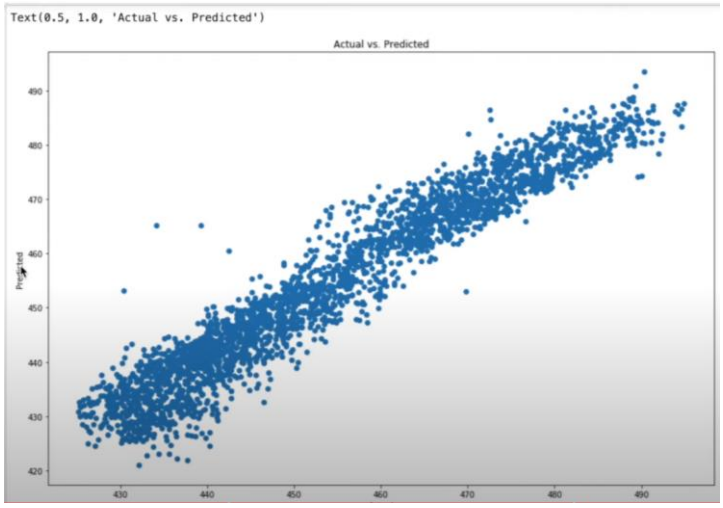
Evaluate the model

```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
```

```
0.9304112159477683
```

Plot the results

```
import matplotlib.pyplot as plt
plt.figure(fisize=(15,10))
plt.scatter(y_test,y_pred)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Actual vs. Predicted')
```



Predicted values

```
pred_y_df=pd.DataFrame({'Actual Value':y_test,'Predicted value':y_pred, 'Difference': y_test-y_pred})
pred_y_df[0:20]
```

	Actual Value	Predicted value	Difference
0	431.23	431.402451	-0.172451
1	460.01	458.614741	1.395259
2	461.14	462.819674	-1.679674
3	445.90	448.601237	-2.701237
4	451.29	457.879479	-6.589479
5	432.68	429.676856	3.003144
6	477.50	473.017115	4.482885
7	459.68	456.532373	3.147627

In [8]:
```
from sklearn.linear_model import LinearRegression
ml=LinearRegression()
ml.fit(x_train,y_train)
```

Out[8]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Predict the test set results

In [9]:
```
y_pred=ml.predict(x_test)
print(y_pred)
```

```
[431.40245096 458.61474119 462.81967423 ... 432.47380825 436.16417243
 439.00714594]
```

In [10]:
```
ml.predict([[14.96,41.76,1024.07,73.17]])
```

Out[10]:
```
array([467.34820092])
```

Evaluate the model

In [ ]:
```
from sklearn.metrics import r2_score
r2_score
```