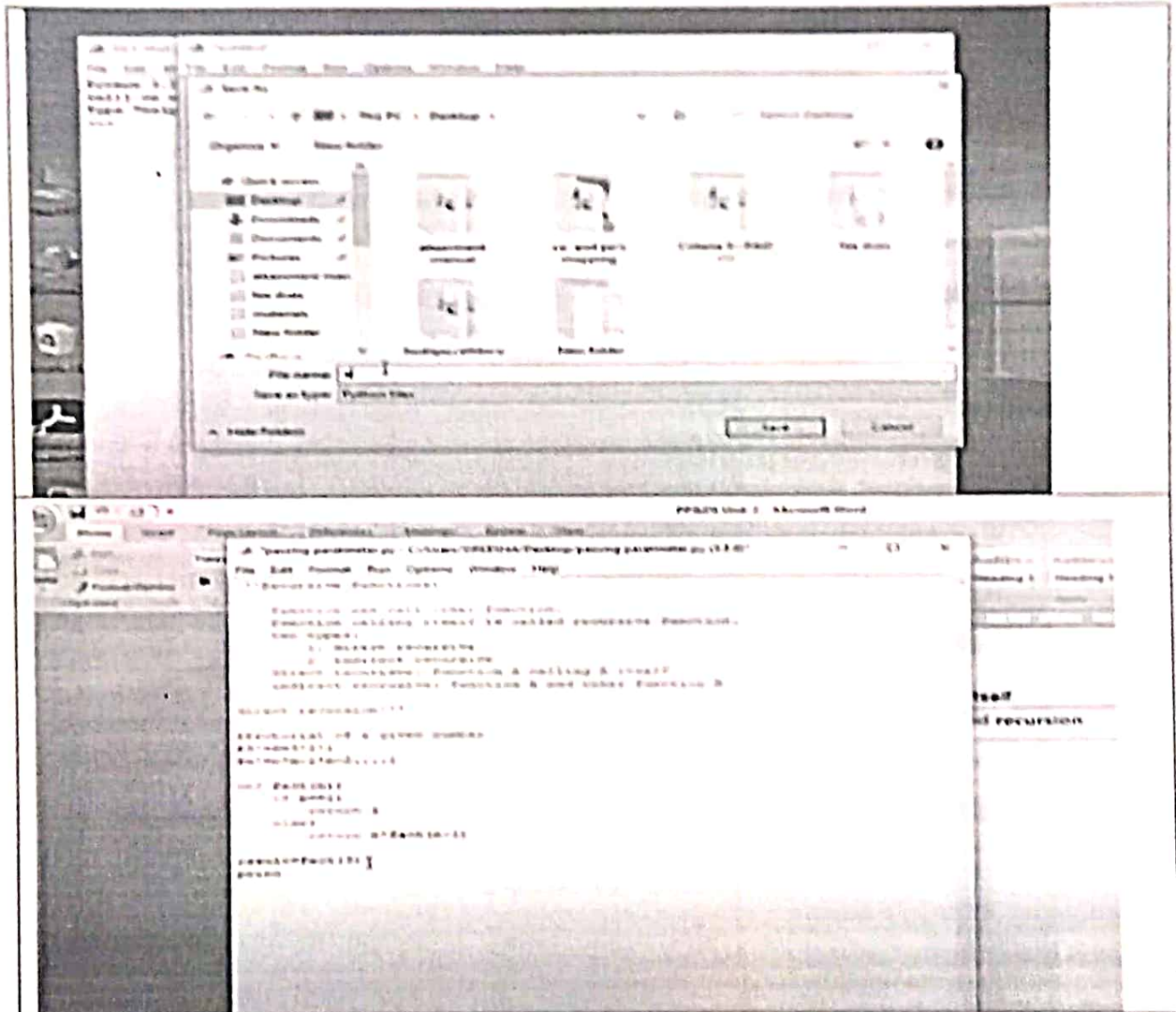




Name of the Subject :	Python Programming	Class:	II Year I Sem
Faculty Name:	Dr.V.Siresha	AY	2020-21

Case studies are stories that are used as a teaching tool to show the application of a theory or concept to real situations. Instructors can create their own cases or can find cases that already exist.



Advantages to the use of case studies in class

A major advantage of teaching with case studies is that the students are actively engaged in figuring out the principles by abstracting from the examples. This develops their skills in:

1. Problem solving
2. Analytical tools, quantitative and/or qualitative, depending on the case
3. Decision making in complex situations.

V. Siresha
Course Instructor

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

+ Code

+ Text

```
data=pd.read_csv("NewspaperData.csv")
data.head()
```

	Newspaper	daily	sunday
0	Baltimore Sun	391.952	488.506
1	Boston Globe	516.981	798.298
2	Boston Herald	355.628	235.084
3	Charlotte Observer	238.555	299.451
4	Chicago Sun Times	537.780	559.093

```
data.shape
```

```
(34, 3)
```

▼ $\text{sunday} = a + b \cdot \text{daily}$

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34 entries, 0 to 33
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Newspaper   34 non-null     object
1   daily       34 non-null     float64
2   sunday      34 non-null     float64
dtypes: float64(2), object(1)
memory usage: 944.0+ bytes
```

```
data.isnull().sum()
```

```
Newspaper    0
daily         0
sunday        0
dtype: int64
```

```
x=data['daily'].values
y=data['sunday'].values
```

```
print(x)
print(y)
```

```
[ 391.952 516.981 355.628 238.555 537.78 733.775 198.832 252.624
 206.204 231.177 449.755 288.571 185.736 1164.388 444.581 412.871
 272.28 781.796 1209.225 825.512 223.748 354.843 515.523 220.465
 337.672 197.12 133.239 374.009 273.844 570.364 391.286 201.86
 321.626 838.902]
[ 488.506 798.298 235.084 299.451 559.093 1133.249 348.744 417.779
 344.522 323.084 620.752 423.305 202.614 1531.527 553.479 685.975
 324.241 983.24 1762.015 960.308 284.611 407.76 982.663 557.
 440.923 268.06 262.048 432.502 338.355 704.322 585.681 267.781
 408.343 1165.567]
```

```
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)
```

```
print(X_train)
print(X_test)
print(y_train)
print(y_test)
```

```
[ 337.672 412.871 206.204 1164.388 273.844 733.775 444.581 570.364
 515.523 516.981 185.736 198.832 220.465 537.78 1209.225 354.843
 825.512 231.177 252.624 321.626 374.009 238.555 391.952]
[288.571 223.748 197.12 781.796 355.628 133.239 201.86 272.28 449.755
 838.902 391.286]
[ 440.923 685.975 344.522 1531.527 338.355 1133.249 553.479 704.322
 982.663 798.298 202.614 348.744 557. 559.093 1762.015 407.76
 960.308 323.084 417.779 408.343 432.502 299.451 488.506]
[ 423.305 284.611 268.06 983.24 235.084 262.048 267.781 324.241
 620.752 1165.567 585.681]
```

```
X_train= X_train.reshape(-1, 1)
y_train= y_train.reshape(-1, 1)
X_test = X_test.reshape(-1, 1)
```

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
```

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

```
# Predicting the Test set results
y_pred = model.predict(X_test)
```

```
print(y_pred)
```

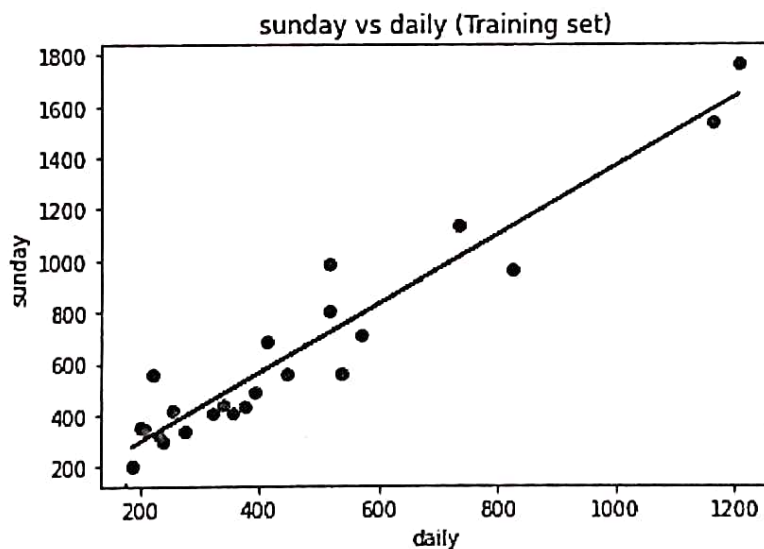
```
[[ 412.69134115]
 [ 326.03682966]
 [ 290.44087866]
 [1072.02786411]
 [ 502.33223368]
 [ 205.04562006]
 [ 296.77724668]
 [ 390.91375223]
```

```
[ 628.15993685]  
[1148.36639415]  
[ 549.99936681]]
```

```
print('Coefficients: \n', model.coef_)  
print('intercept:', model.intercept_)
```

```
Coefficients:  
[[1.3367865]]  
intercept: [26.93352318]
```

```
plt.scatter(X_train, y_train, color = 'red')  
plt.plot(X_train, model.predict(X_train), color = 'blue')  
plt.title('sunday vs daily (Training set)')  
plt.xlabel('daily')  
plt.ylabel('sunday')  
plt.show()
```



```
# Visualising the Test set results  
plt.scatter(X_test, y_test, color = 'red')  
plt.plot(X_train, model.predict(X_train), color = 'blue')  
plt.title('sunday vs daily (Test set)')  
plt.xlabel('daily')  
plt.ylabel('Sunday')  
plt.show()
```

