Problem Statement 1

In [19]:

```
import pandas as pd
data = pd.read_csv("C:/Users/Hp/Downloads/delivery_time.csv")
data
```

Out[19]:

	Delivery Time	Sorting Time
0	21.00	10
1	13.50	4
2	19.75	6
3	24.00	9
4	29.00	10
5	15.35	6
6	19.00	7
7	9.50	3
8	17.90	10
9	18.75	9
10	19.83	8
11	10.75	4
12	16.68	7
13	11.50	3
14	12.03	3
15	14.88	4
16	13.75	6
17	18.11	7
18	8.00	2
19	17.83	7
20	21.50	5

In [4]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 2 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Delivery Time  21 non-null     float64
 1   Sorting Time   21 non-null     int64  
dtypes: float64(1), int64(1)
memory usage: 464.0 bytes
```

In [5]:

```
data.corr()
```

Out[5]:

	Delivery Time	Sorting Time
Delivery Time	1.000000	0.825997
Sorting Time	0.825997	1.000000

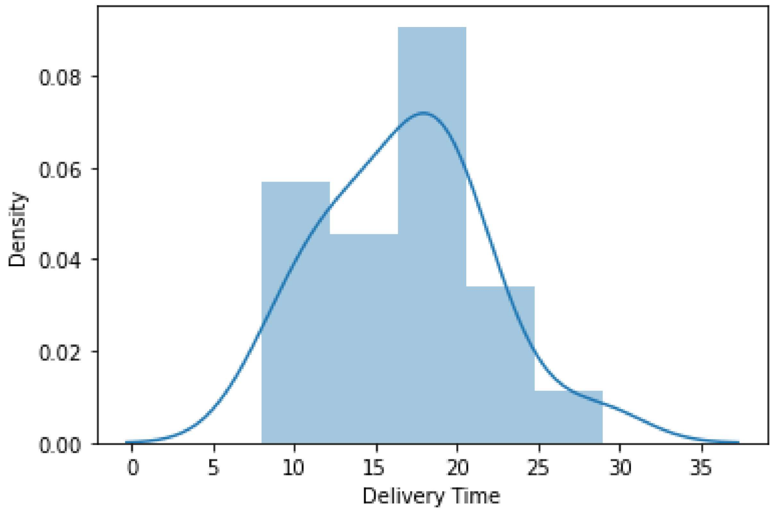
In [7]:

```
import seaborn as sns
sns.distplot(data['Delivery Time'])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Futur
eWarning: `distplot` is a deprecated function and will be removed in a futur
e version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```

Out[7]:

```
<AxesSubplot:xlabel='Delivery Time', ylabel='Density'>
```



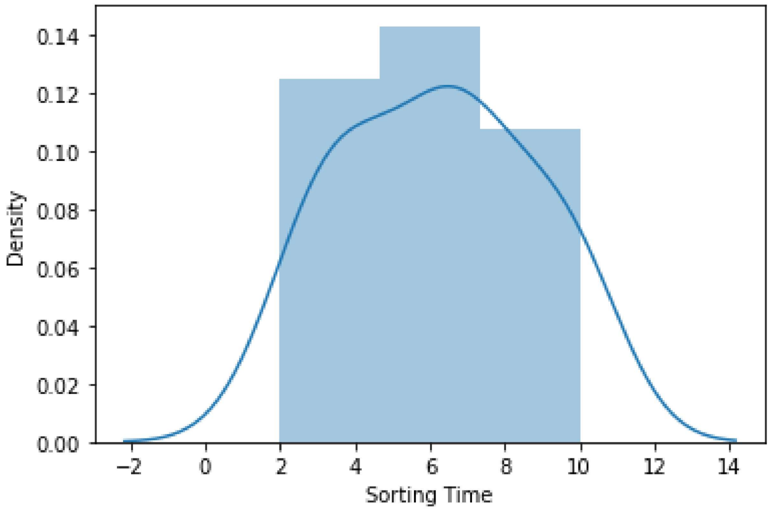
In [8]:

```
import seaborn as sns
sns.distplot(data['Sorting Time'])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Futur
eWarning: `distplot` is a deprecated function and will be removed in a futur
e version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```

Out[8]:

```
<AxesSubplot:xlabel='Sorting Time', ylabel='Density'>
```



In [9]:

```
import statsmodels.formula.api as smf
```

In [10]:

```
data=data.rename({'Delivery Time':'Delivery_Time','Sorting Time':'Sorting_Time'},axis=1)
```

In [11]:

```
data
```

Out[11]:

	Delivery_Time	Sorting_Time
0	21.00	10
1	13.50	4
2	19.75	6
3	24.00	9
4	29.00	10
5	15.35	6
6	19.00	7
7	9.50	3
8	17.90	10
9	18.75	9
10	19.83	8
11	10.75	4
12	16.68	7
13	11.50	3
14	12.03	3
15	14.88	4
16	13.75	6
17	18.11	7
18	8.00	2
19	17.83	7
20	21.50	5

In [12]:

```
model=smf.ols("Delivery_Time~Sorting_Time", data=data).fit()
```

In [13]:

```
model.params
```

Out[13]:

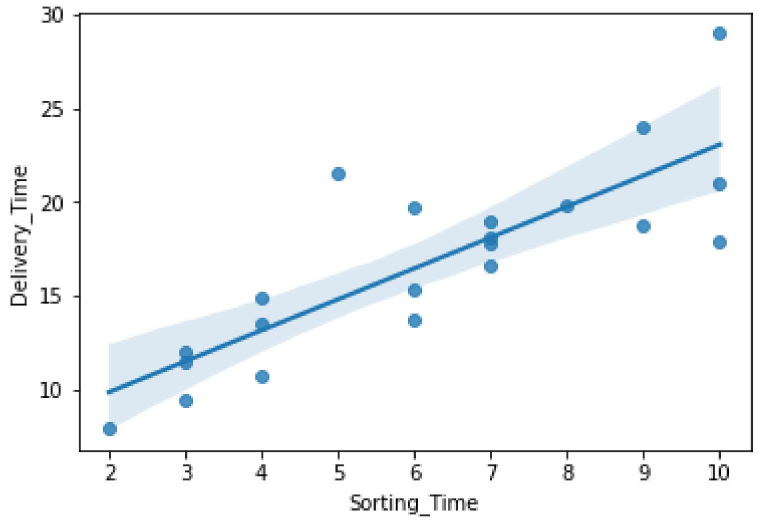
```
Intercept       6.582734
Sorting_Time    1.649020
dtype: float64
```

In [14]:

```
sns.regplot(x='Sorting_Time',y='Delivery_Time',data=data)
```

Out[14]:

```
<AxesSubplot:xlabel='Sorting_Time', ylabel='Delivery_Time'>
```



In [15]:

```
print(model.tvalues,'\n',model.pvalues)
```

```
Intercept       3.823349
Sorting_Time    6.387447
dtype: float64
 Intercept       0.001147
Sorting_Time    0.000004
dtype: float64
```

In [16]:

```
(model.rsquared,model.rsquared_adj)
```

Out[16]:

```
(0.6822714748417231, 0.6655489208860244)
```

In [17]:

```
newdata=pd.Series([2,4,6,8,10])
```

In [18]:

```
newdata
```

Out[18]:

```
0     2
1     4
2     6
3     8
4    10
dtype: int64
```

In [19]:

```
data_pred=pd.DataFrame(newdata,columns=['Sorting_Time'])
```

In [20]:

```
model.predict(data_pred)
```

Out[20]:

```
0     9.880774
1    13.178814
2    16.476853
3    19.774893
4    23.072933
dtype: float64
```

Problem statement 2

In [1]:

```
import pandas as pd
data = pd.read_csv("C:/Users/Hp/Downloads/Salary_Data.csv")
data
```

Out[1]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0
10	3.9	63218.0
11	4.0	55794.0
12	4.0	56957.0
13	4.1	57081.0
14	4.5	61111.0
15	4.9	67938.0
16	5.1	66029.0
17	5.3	83088.0
18	5.9	81363.0
19	6.0	93940.0
20	6.8	91738.0
21	7.1	98273.0
22	7.9	101302.0
23	8.2	113812.0
24	8.7	109431.0
25	9.0	105582.0
26	9.5	116969.0
27	9.6	112635.0
28	10.3	122391.0
29	10.5	121872.0

In [3]:

```
data.corr()
```

Out[3]:

	YearsExperience	Salary
YearsExperience	1.000000	0.978242
Salary	0.978242	1.000000

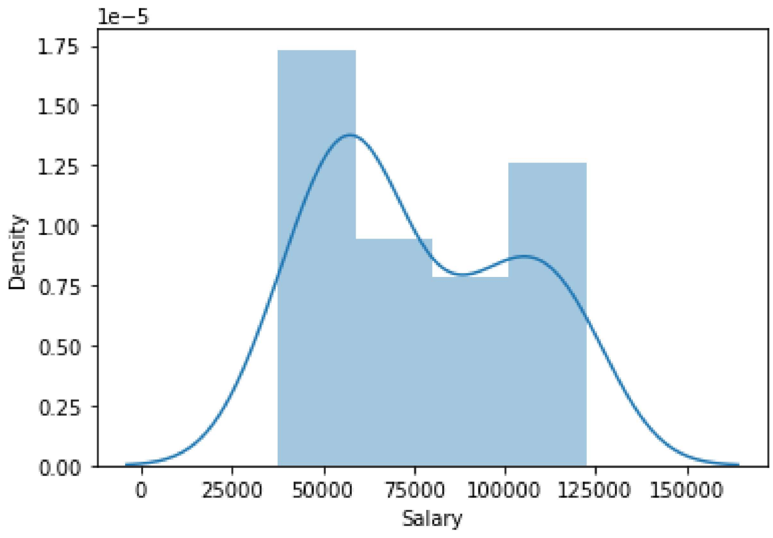
In [4]:

```
import seaborn as sns
sns.distplot(data['Salary'])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Futur
eWarning: `distplot` is a deprecated function and will be removed in a futur
e version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```

Out[4]:

```
<AxesSubplot:xlabel='Salary', ylabel='Density'>
```



In [10]:

```
import statsmodels.formula.api as smf
model = smf.ols("Salary~YearsExperience",data = data).fit()
```

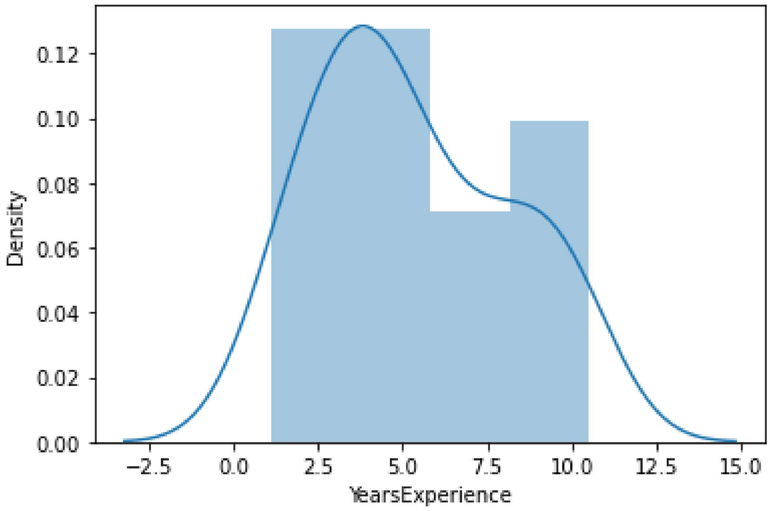
In [5]:

```
sns.distplot(data['YearsExperience'])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Futur
eWarning: `distplot` is a deprecated function and will be removed in a futur
e version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for h
istograms).
  warnings.warn(msg, FutureWarning)
```

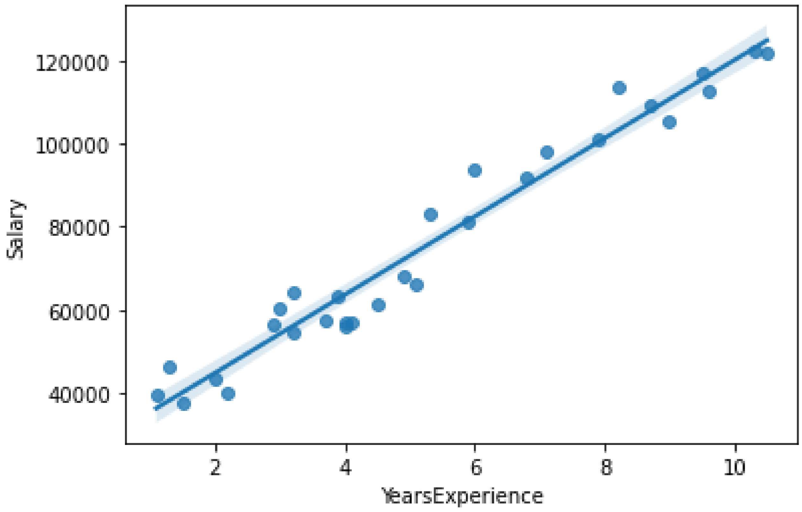
Out[5]:

```
<AxesSubplot:xlabel='YearsExperience', ylabel='Density'>
```



In [7]:

```
sns.regplot(x="YearsExperience",y="Salary",data=data);
```



In [11]:

```
model=smf.ols("Salary~YearsExperience",data=data).fit()
```

In [12]:

```
model.params
```

Out[12]:

```
Intercept          25792.200199
YearsExperience     9449.962321
dtype: float64
```

In [13]:

```
print(model.tvalues,'\n',model.pvalues)
```

```
Intercept          11.346940
YearsExperience    24.950094
dtype: float64
 Intercept          5.511950e-12
YearsExperience    1.143068e-20
dtype: float64
```

In [14]:

```
(model.rsquared,model.rsquared_adj)
```

Out[14]:

```
(0.9569566641435086, 0.9554194021486339)
```

In [15]:

```
(model.rsquared,model.rsquared_adj)
```

Out[15]:

```
(0.9569566641435086, 0.9554194021486339)
```

In [16]:

```
newdata=pd.Series([5,10,15,20])
```

In [17]:

```
data_pred=pd.DataFrame(newdata,columns=['YearsExperience'])
```

In [18]:

```
model.predict(data_pred)
```

Out[18]:

```
0     73042.011806
1    120291.823413
2    167541.635020
3    214791.446628
dtype: float64
```

In [ ]:

```
```