```
In [1]: pip install linear regression
```

Requirement already satisfied: linear in c:\users\dinesh reddy\appdata\local
\programs\python\python310\lib\site-packages (0.0.dev0)
Collecting regression
  Using cached regression-1.0.5-py3-none-any.whl (17 kB)
Collecting click==8.1.3 (from regression)
  Using cached click-8.1.3-py3-none-any.whl (96 kB)
Collecting colorama==0.4.5 (from regression)
  Using cached colorama-0.4.5-py2.py3-none-any.whl (16 kB)
Requirement already satisfied: numpy==1.23.3 in c:\users\dinesh reddy\appdat
a\local\programs\python\python310\lib\site-packages (from regression) (1.23.
3)
Collecting optimisation-algorithms==1.1.2 (from regression)
  Using cached optimisation_algorithms-1.1.2-py3-none-any.whl (26 kB)
Installing collected packages: colorama, click, optimisation-algorithms, reg
ression
  Attempting uninstall: colorama
    Found existing installation: colorama 0.4.6
    Uninstalling colorama-0.4.6:
      Successfully uninstalled colorama-0.4.6
Successfully installed click-8.1.3 colorama-0.4.5 optimisation-algorithms-1.
1.2 regression-1.0.5
Note: you may need to restart the kernel to use updated packages.

```
In [16]: import numpy as np
         import pandas as pd
         import seaborn as sns
         import matplotlib.pyplot as plt
         from sklearn import preprocessing,svm
         from sklearn.model_selection import train_test_split
         from sklearn.linear_model import LinearRegression
```

```
In [20]: df=pd.read_csv(r"C:\Users\dinesh reddy\AppData\Local\Temp\Temp1_bottle.csv.zip
         df
```

C:\Users\dinesh reddy\AppData\Local\Temp\ipykernel_7208\1685568230.py:1: Dty
peWarning: Columns (47,73) have mixed types. Specify dtype option on import
or set low_memory=False.
   df=pd.read_csv(r"C:\Users\dinesh reddy\AppData\Local\Temp\Temp1_bottle.cs
v.zip\bottle.csv")

Out[20]:

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2S
0	1	1	054.0 056.0	19-4903CR-HY-060-0930-05400560-0000A-3	0	10.500	33.4400	NaN	25.64900	N
1	1	2	054.0 056.0	19-4903CR-HY-060-0930-05400560-0008A-3	8	10.460	33.4400	NaN	25.65600	N
2	1	3	054.0 056.0	19-4903CR-HY-060-0930-05400560-0010A-7	10	10.460	33.4370	NaN	25.65400	N
3	1	4	054.0 056.0	19-4903CR-HY-060-0930-05400560-0019A-3	19	10.450	33.4200	NaN	25.64300	N
4	1	5	054.0 056.0	19-4903CR-HY-060-0930-05400560-0020A-7	20	10.450	33.4210	NaN	25.64300	N
...	...	...	...	...	...	...	...	...	...	
864858	34404	864859	093.4 026.4	20-1611SR-MX-310-2239-09340264-0000A-7	0	18.744	33.4083	5.805	23.87055	108
864859	34404	864860	093.4 026.4	20-1611SR-MX-310-2239-09340264-0002A-3	2	18.744	33.4083	5.805	23.87072	108
864860	34404	864861	093.4 026.4	20-1611SR-MX-310-2239-09340264-0005A-3	5	18.692	33.4150	5.796	23.88911	108
864861	34404	864862	093.4 026.4	20-1611SR-MX-310-2239-09340264-0010A-3	10	18.161	33.4062	5.816	24.01426	107

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2S
864862	34404	864863	093.4 026.4	20-1611SR-MX-310-2239-09340264-0015A-3	15	17.533	33.3880	5.774	24.15297	105

864863 rows × 74 columns

```
In [21]: df.head(10)
```

Out[21]:

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	...	
0	1	1	054.0 056.0	19-4903CR-HY-060-0930-05400560-0000A-3	0	10.50	33.440	NaN	25.649	NaN	...	
1	1	2	054.0 056.0	19-4903CR-HY-060-0930-05400560-0008A-3	8	10.46	33.440	NaN	25.656	NaN	...	
2	1	3	054.0 056.0	19-4903CR-HY-060-0930-05400560-0010A-7	10	10.46	33.437	NaN	25.654	NaN	...	
3	1	4	054.0 056.0	19-4903CR-HY-060-0930-05400560-0019A-3	19	10.45	33.420	NaN	25.643	NaN	...	
4	1	5	054.0 056.0	19-4903CR-HY-060-0930-05400560-0020A-7	20	10.45	33.421	NaN	25.643	NaN	...	
5	1	6	054.0 056.0	19-4903CR-HY-060-0930-05400560-0030A-7	30	10.45	33.431	NaN	25.651	NaN	...	
6	1	7	054.0 056.0	19-4903CR-HY-060-0930-05400560-0039A-3	39	10.45	33.440	NaN	25.658	NaN	...	
7	1	8	054.0 056.0	19-4903CR-HY-060-0930-05400560-0050A-7	50	10.24	33.424	NaN	25.682	NaN	...	
8	1	9	054.0 056.0	19-4903CR-HY-060-0930-05400560-0058A-3	58	10.06	33.420	NaN	25.710	NaN	...	

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	...	
9	1	10	054.0 056.0	19-4903CR-HY-060-0930-05400560-0075A-7	75	9.86	33.494	NaN	25.801	NaN	...	

10 rows × 74 columns

In [28]:
```python
df=df[['Salnty','T_degC']]
df.columns=['Sal','Temp']
df.head(10)
```

Out[28]:

	Sal	Temp
0	33.440	10.50
1	33.440	10.46
2	33.437	10.46
3	33.420	10.45
4	33.421	10.45
5	33.431	10.45
6	33.440	10.45
7	33.424	10.24
8	33.420	10.06
9	33.494	9.86

```
In [31]: sns.lmplot(x="Sal",y="Temp",data=df,order=2,ci=None)
```

Out[31]: <seaborn.axisgrid.FacetGrid at 0x16595277c70>
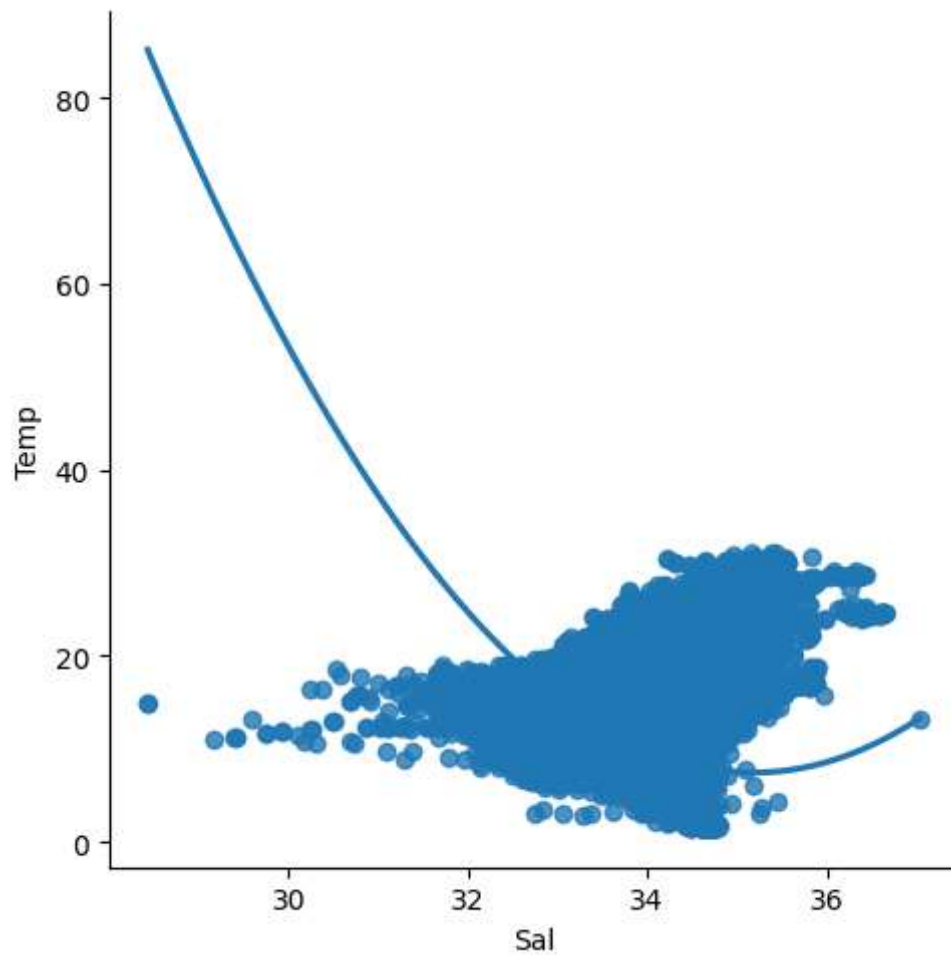

```
In [32]: df.describe()
```

Out[32]:

	Sal	Temp
count	817509.000000	853900.000000
mean	33.840350	10.799677
std	0.461843	4.243825
min	28.431000	1.440000
25%	33.488000	7.680000
50%	33.863000	10.060000
75%	34.196900	13.880000
max	37.034000	31.140000

```
In [34]: df.fillna(method="ffill",inplace=True)
```

C:\Users\dinesh reddy\AppData\Local\Temp\ipykernel_7208\1844562654.py:1: Set
tingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  df.fillna(method="ffill",inplace=True)

```
In [36]: x=np.array(df['Sal']).reshape(-1,1)
         y=np.array(df['Temp']).reshape(-1,1)
```

```
In [37]: df.dropna(inplace=True)
```

C:\Users\dinesh reddy\AppData\Local\Temp\ipykernel_7208\1379821321.py:1: Set
tingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  df.dropna(inplace=True)

```
In [43]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
         regr=LinearRegression()
         regr.fit(x_train,y_train)
         print(regr.score(x_test,y_test))
```
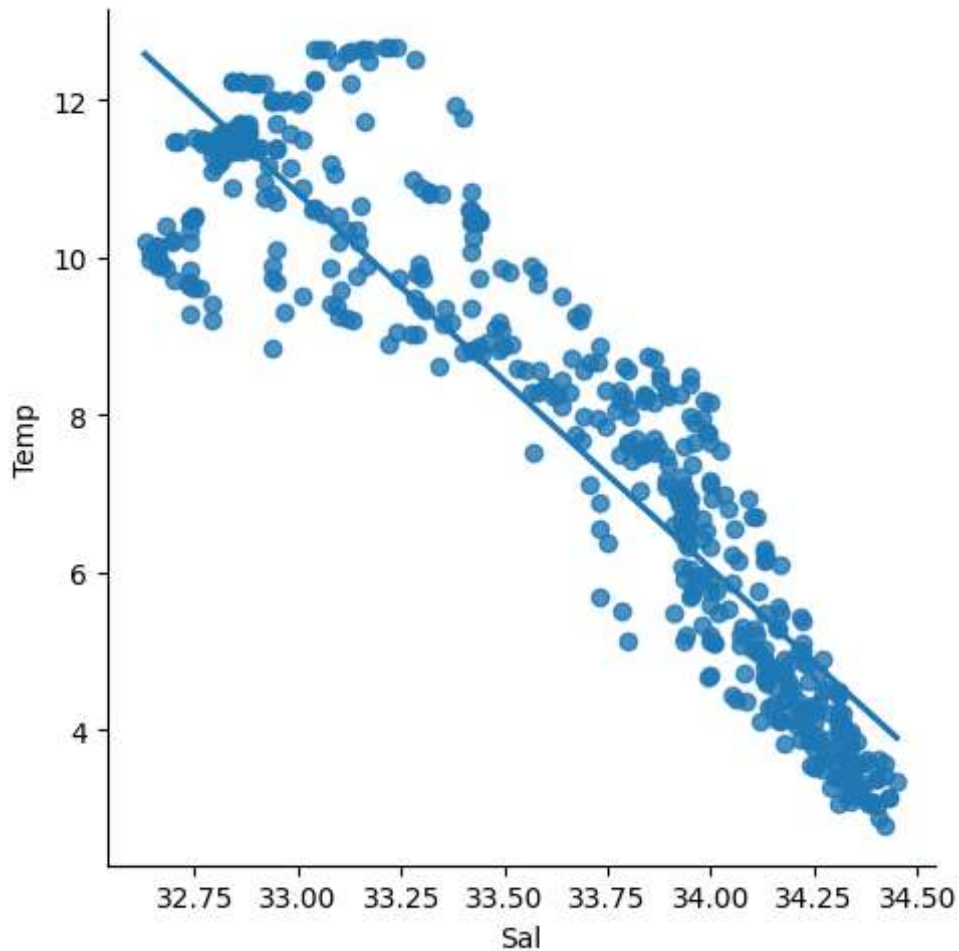
0.2037749722375608

```
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```
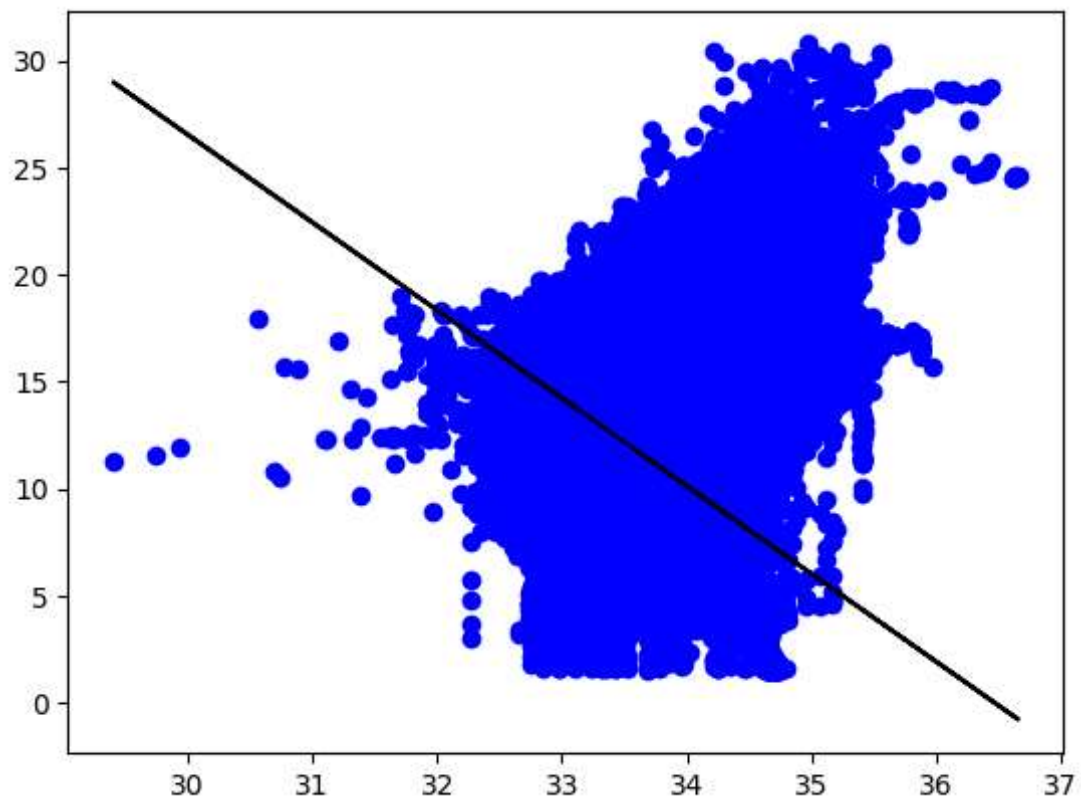
```
In [47]: df500=df[:][:500]
         sns.lmplot(x="Sal",y="Temp",data=df500,order=1,ci=None)
```

Out[47]: <seaborn.axisgrid.FacetGrid at 0x16599ccb490>


```
In [48]: df500.fillna(method='ffill',inplace=True)
```

```
In [49]: x=np.array(df500['Sal']).reshape(-1,1)
         y=np.array(df500['Temp']).reshape(-1,1)
```
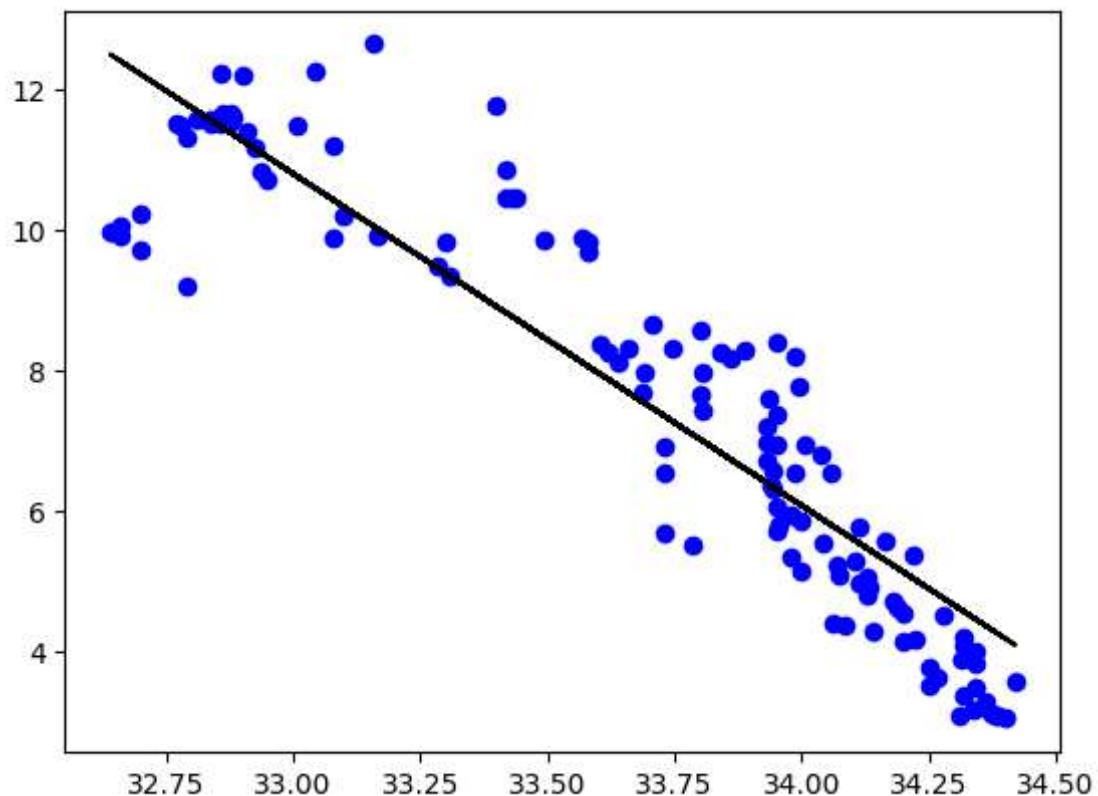
```
In [50]: df500.dropna(inplace=True)
```

```
In [51]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
```

```
In [52]: regr=LinearRegression()
         regr.fit(x_train,y_train)
         print("Regression:",regr.score(x_test,y_test))
```

         Regression: 0.8512810988624531

In [53]:
```python
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```


In [54]:
```python
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
model=LinearRegression()
model.fit(x_train,y_train)
```

Out[54]:
```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.

On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [55]:
```python
y_pred=model.predict(x_test)
r2=r2_score(y_test,y_pred)
print("R2 score:",r2)
```

```
R2 score: 0.8512810988624531
```

In [ ]: