In [2]:

```
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 [3]:

```
#Step-2:Reading the dataset
df=pd.read_csv(r"C:\Users\91955\Desktop\Data Analysis with Python\bottle.csv")
df
```

```
C:\Users\91955\AppData\Local\Temp\ipykernel_2828\292355879.py:2: DtypeWarn
ing: Columns (47,73) have mixed types. Specify dtype option on import or s
et low_memory=False.
  df=pd.read_csv(r"C:\Users\91955\Desktop\Data Analysis with Python\bottl
e.csv")
```

Out[3]:

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

In [4]:

```
df=df[['Salnty','T_degC']]
#Taking only the selected two attributes from the dataset
df.columns=['Sal','Temp']
#Renaming the columns for easier writing of the code
```

In [5]:

```
df.head(10)
#Displaying only the 1st 10 rows
```

Out[5]:

	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

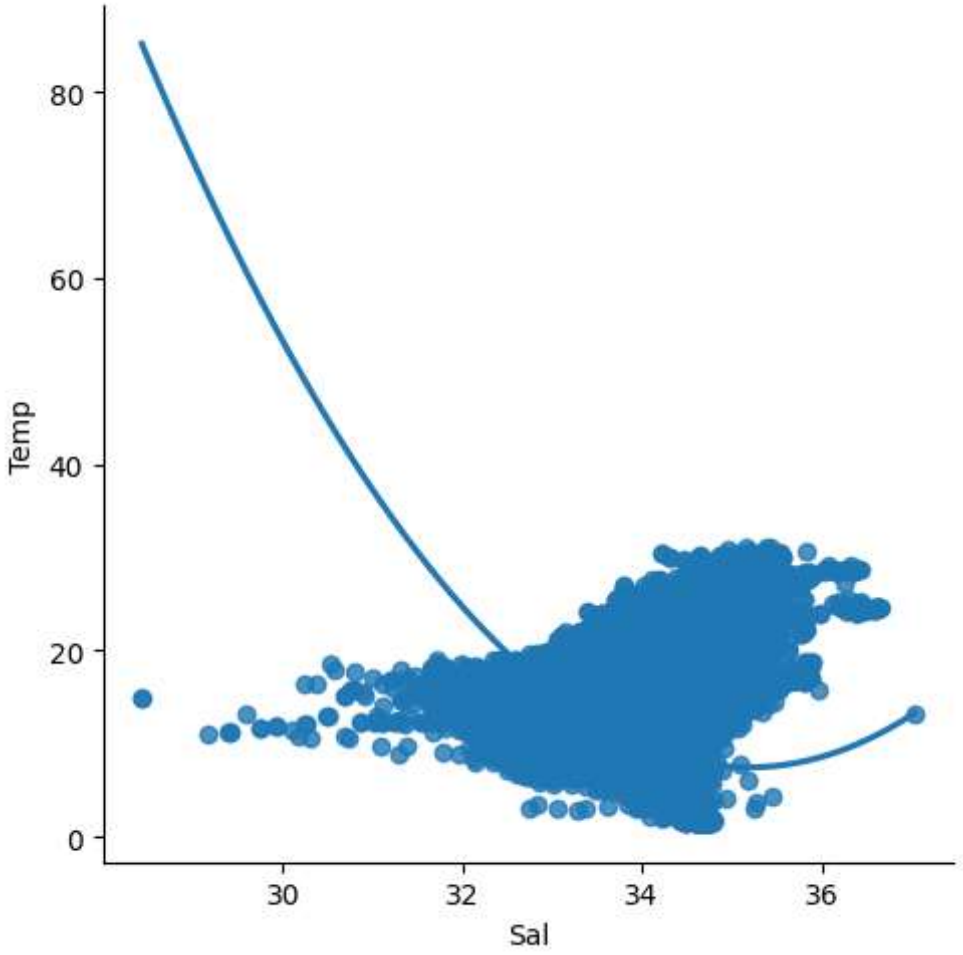
In [24]:

```
sns.lmplot(x='Sal',y='Temp',data=df,order=2,ci=None)
```

Out[24]:

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

```
<seaborn.axisgrid.FacetGrid at 0x276a76a4580>
```



In [25]:

```
df.describe()
```

Out[25]:

	Sal	Temp
count	814247.000000	814247.000000
mean	33.841337	10.860287
std	0.461636	4.224930
min	28.431000	1.440000
25%	33.489000	7.750000
50%	33.866000	10.110000
75%	34.197000	13.930000
max	37.034000	31.140000

In [26]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 814247 entries, 0 to 864862
Data columns (total 2 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   Sal     814247 non-null  float64
 1   Temp    814247 non-null  float64
dtypes: float64(2)
memory usage: 18.6 MB
```

In [27]:

```
df.fillna(method='ffill')
```

Out[27]:

	Sal	Temp
0	33.4400	10.500
1	33.4400	10.460
2	33.4370	10.460
3	33.4200	10.450
4	33.4210	10.450
...	...	...
864858	33.4083	18.744
864859	33.4083	18.744
864860	33.4150	18.692
864861	33.4062	18.161
864862	33.3880	17.533

814247 rows × 2 columns

In [29]:

```
x=np.array(df['Sal']).reshape(-1,1)
```

In [30]:

```
y=np.array(df['Temp']).reshape(-1,1)
```

In [31]:

```
df.dropna(inplace=True)
```

```
C:\Users\91955\AppData\Local\Temp\ipykernel_2828\1379821321.py:1: SettingW
ithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://
pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
view-versus-a-copy)
  df.dropna(inplace=True)
```

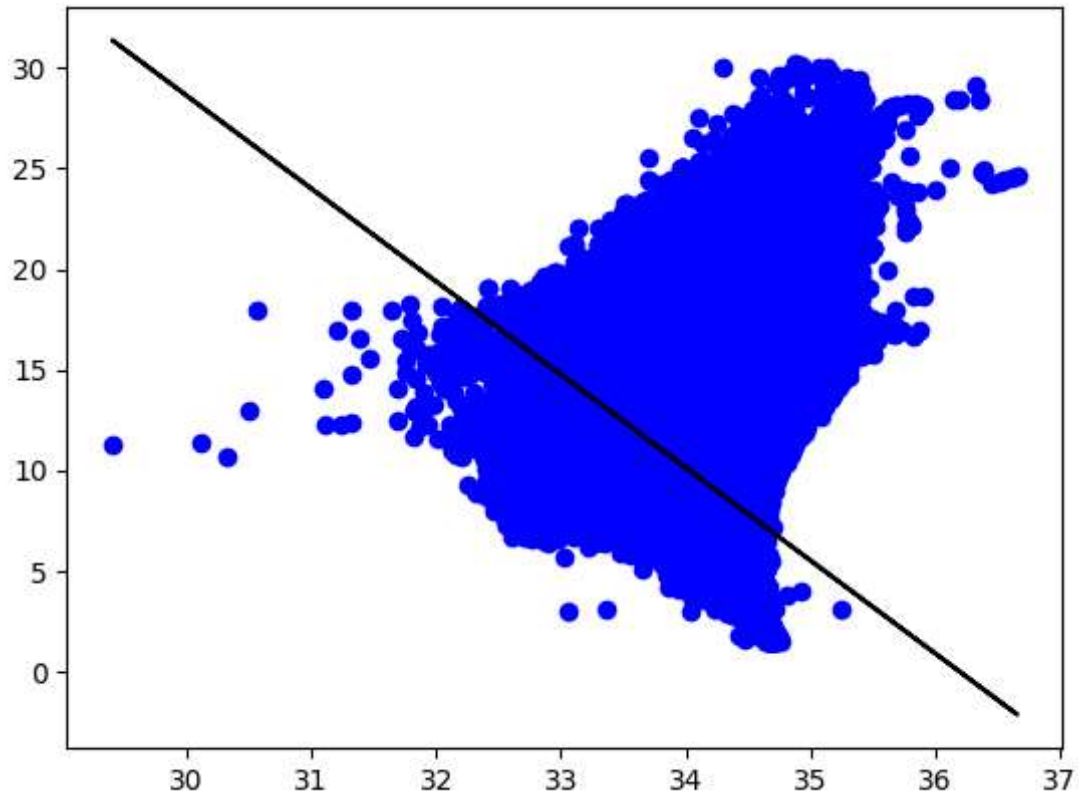
In [41]:

```
X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
reg=LinearRegression()q
reg.fit(X_train,y_train)
print(reg.score(X_test,y_test))
```

```
0.2544304702117517
```

In [43]:

```
y_pred=reg.predict(X_test)
plt.scatter(X_test,y_test,color='b')
plt.plot(X_test,y_pred,color='k')
plt.show()
```

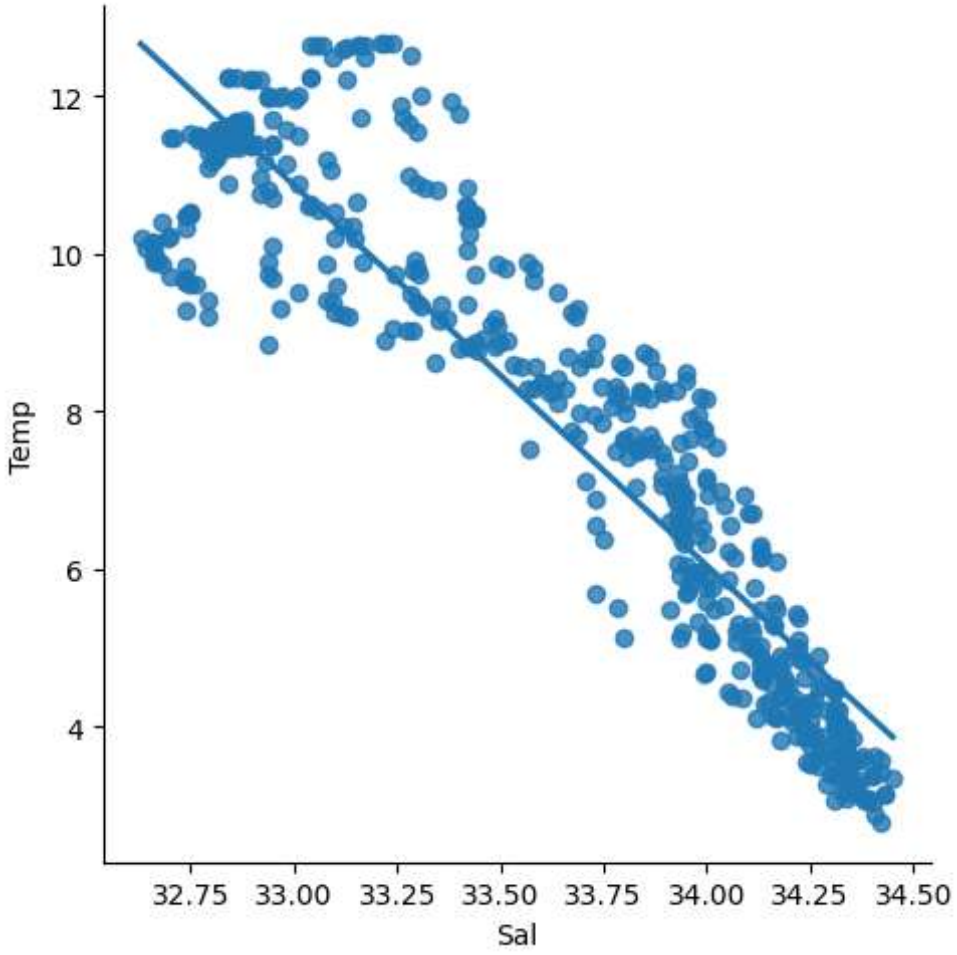


In [44]:

```
df500=df[:][:500]
sns.lmplot(x='Sal',y='Temp',data=df500,order=1,ci=None)
```

Out[44]:

```
<seaborn.axisgrid.FacetGrid at 0x276a7840cd0>
```



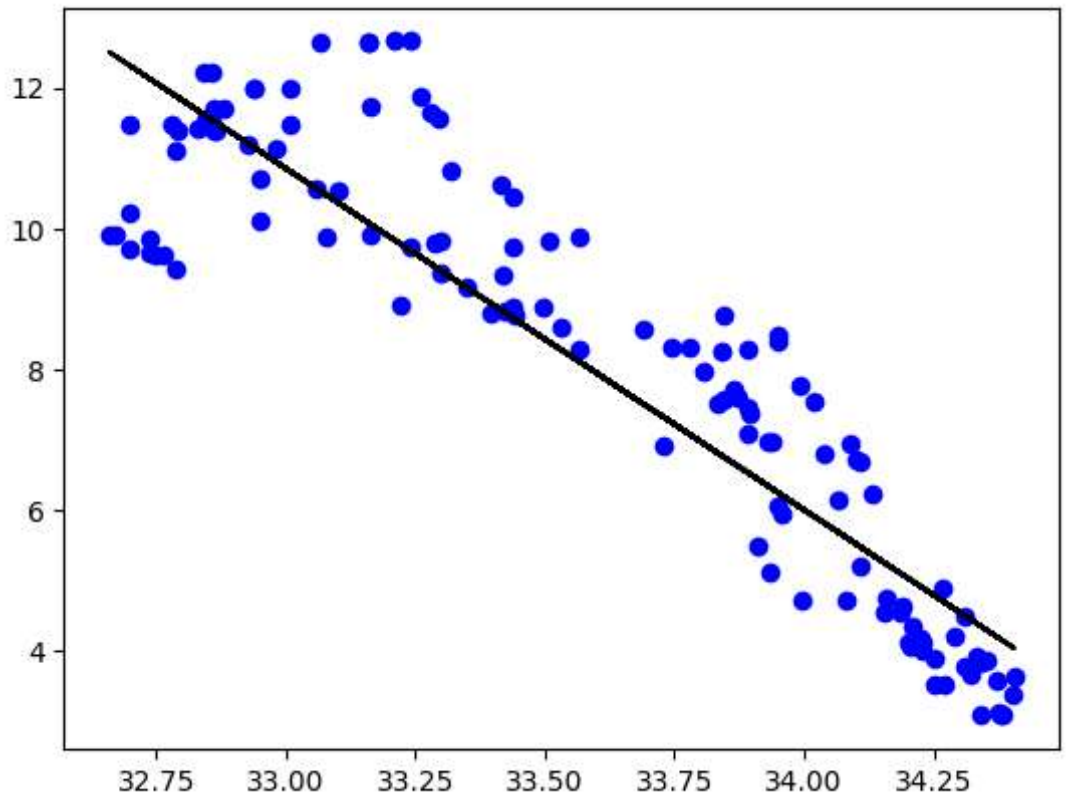
In [46]:

```
df500.fillna(method='ffill',inplace=True)
X=np.array(df500['Sal']).reshape(-1,1)
y=np.array(df500['Temp']).reshape(-1,1)
df500.dropna(inplace=True)
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25)
reg=LinearRegression()
reg.fit(X_train,y_train)
print("Regression:",reg.score(X_test,y_test))
y_pred=reg.predict(X_test)
plt.scatter(X_test,y_test,color='b')
plt.plot(X_test,y_pred,color='k')
plt.show
```

```
Regression: 0.8180243753722324
```

Out[46]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [48]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
mode1=LinearRegression()
mode1.fit(X_train,y_train)
y_pred=mode1.predict(X_test)
r2=r2_score(y_test,y_pred)
print("R2 score: ",r2)
```

```
R2 score:  0.8180243753722324
```

In [ ]:

```
1 #conclusion:Linear regression is the best fit for the model
```