In [1]:
```
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 [2]:
```
df=pd.read_csv(r"C:\Users\LENOVO\Desktop\bottle.csv")
df
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_8420\496985732.py:1: DtypeWarn
ing: Columns (47,73) have mixed types. Specify dtype option on import or se
t low_memory=False.
  df=pd.read_csv(r"C:\Users\LENOVO\Desktop\bottle.csv")
```

Out[2]:

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O
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-	8	10.460	33.4400	NaN	25.65600	

In [3]:
```
df.shape
```

Out[3]:
```
(864863, 74)
```

In [4]:
```
df=df[['Salnty','T_degC']]
df.columns=['Sal','Temp']
```

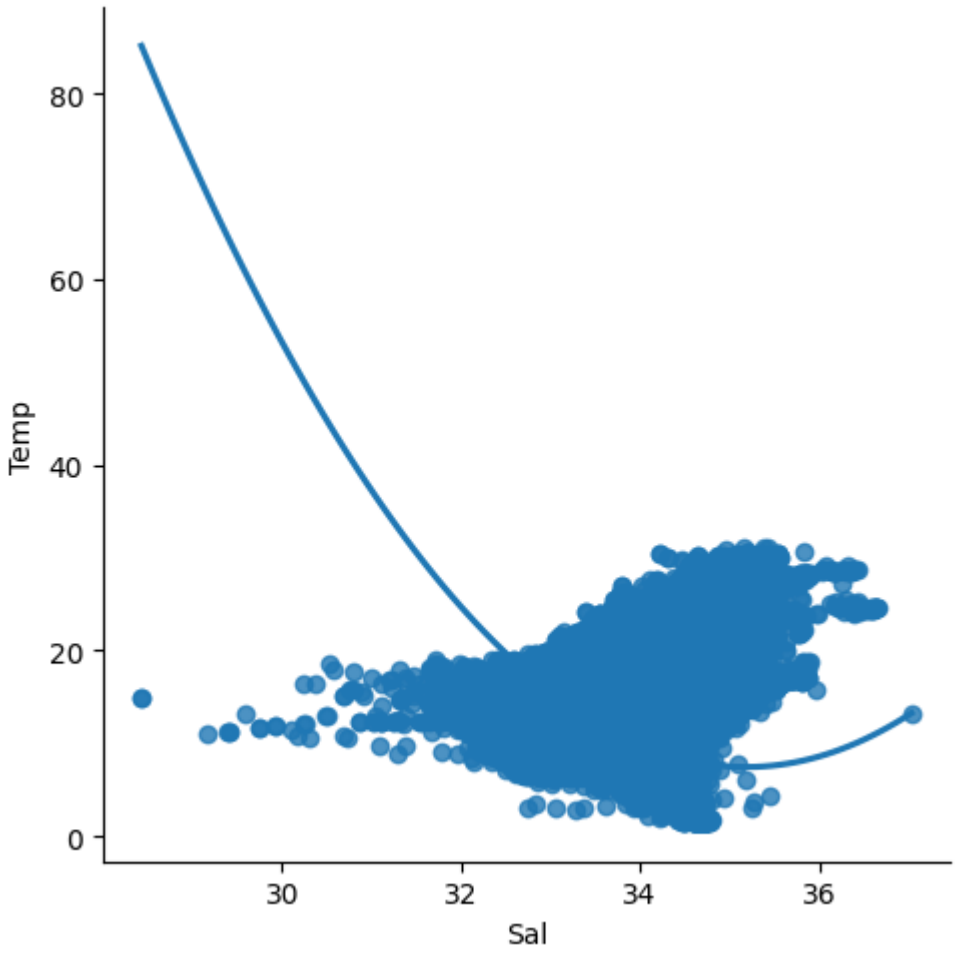
In [5]:
```
df.head(10)
```

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 [6]:
```
sns.lmplot(x="Sal",y="Temp",data=df,order=2,ci=None)
```

Out[6]:
```
<seaborn.axisgrid.FacetGrid at 0x2bcc7668610>
```



In [7]:
```
df.describe()
```

Out[7]:

	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 [8]:
```
df.fillna(method='ffill',inplace=True)
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_8420\4116506308.py:1: SettingWit
hCopyWarning:
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/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  df.fillna(method='ffill',inplace=True)
```

In [9]:
```
x=np.array(df['Sal']).reshape(-1,1)
y=np.array(df['Temp']).reshape(-1,1)
```

In [10]:
```
df.dropna(inplace=True)
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_8420\1379821321.py:1: SettingWit
hCopyWarning:
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/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  df.dropna(inplace=True)
```

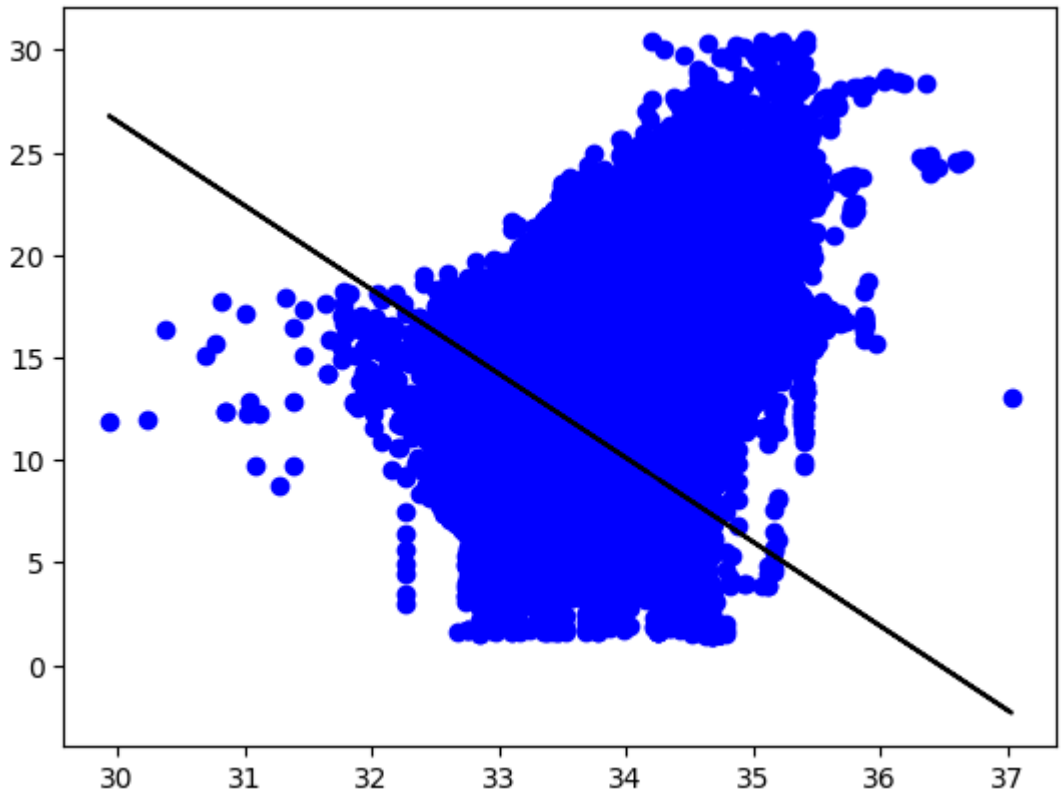
In [16]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
```

In [17]:
```
regr=LinearRegression()
```

In [18]:
```
regr.fit(x_train,y_train)
print(regr.score(x_test,y_test))
```

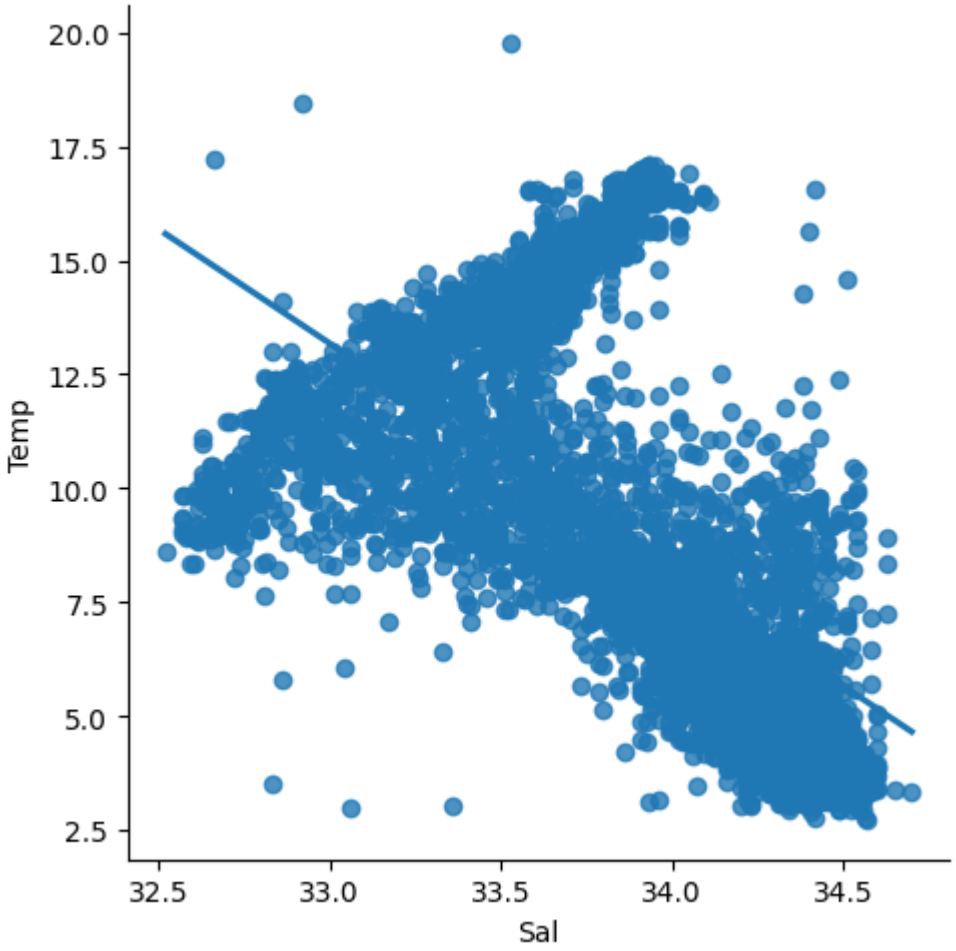
```
0.20499305717069405
```

In [21]:
```
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```



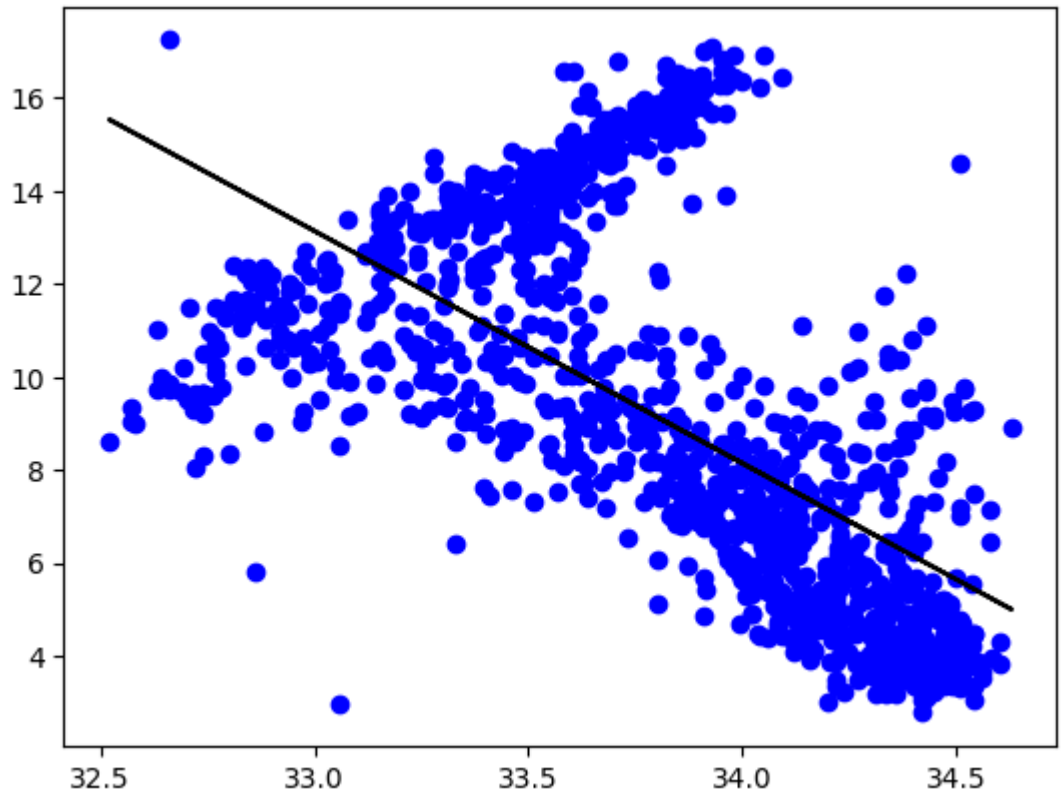
In [23]:
```
df500=df[:][:5000]
sns.lmplot(x="Sal",y="Temp",data=df500,order=1,ci=None)
```

Out[23]:
```
<seaborn.axisgrid.FacetGrid at 0x2bcd89be5d0>
```



In [26]:
```
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)
a=LinearRegression()
a.fit(x_train,y_train)
print(a.score(x_test,y_test))
y_pred=a.predict(x_test)
plt.scatter(x_test,y_test,color="b")
plt.plot(x_test,y_pred,color="k")
plt.show()
```

```
0.4033987216832552
```



In [34]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
model = LinearRegression()
model.fit(x_train,y_train)
```

Out[34]:
```
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 [35]:
```
y_pred=model.predict(x_test)
r2=r2_score(y_test,y_pred)
print("R2 score:",r2)
```

```
R2 score: 0.4033987216832552
```

In [ ]:
```

```