Relationship between salinity & water temp.(ML)

In [5]:

```
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 [7]:

```
df=pd.read_csv(r"C:\Users\prajapath Arjun\Downloads\bottle.csv\bottle.csv")
df
```

```
C:\Users\prajapath Arjun\AppData\Local\Temp\ipykernel_25184\930288583.py:
1: DtypeWarning: Columns (47,73) have mixed types. Specify dtype option on
import or set low_memory=False.
  df=pd.read_csv(r"C:\Users\prajapath Arjun\Downloads\bottle.csv\bottle.cs
v")
```

Out[7]:

	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 [11]:

```
df=df[['Salnty','T_degC']]
```

In [12]:

```
df.columns=['sal','Temp']
```

In [14]:

```
df.head(10)
```

Out[14]:

	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 [18]:

```
df.isna().any()
```

Out[18]:

```
sal     True
Temp    True
dtype: bool
```

In [19]:

```
df
```

Out[19]:

	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

	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

864863 rows × 74 columns

864863 rows × 2 columns

In [16]:

```
df1=df.dropna()
```

In [17]:

```
df1
```

Out[17]:

	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 [20]:

```
df
```

Out[20]:

	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

864863 rows × 2 columns

In [42]:

```
fill_null=["sal","Temp"]
for column in fill_null:
    mean=df[column].mean()
    df[column].fillna(mean,inplace=True)
```

```
C:\Users\prajapath Arjun\AppData\Local\Temp\ipykernel_25184\1742398843.py:
4: SettingWithCopyWarning:
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[column].fillna(mean,inplace=True)
```

In [43]:

```
df.isna().any()
```

Out[43]:

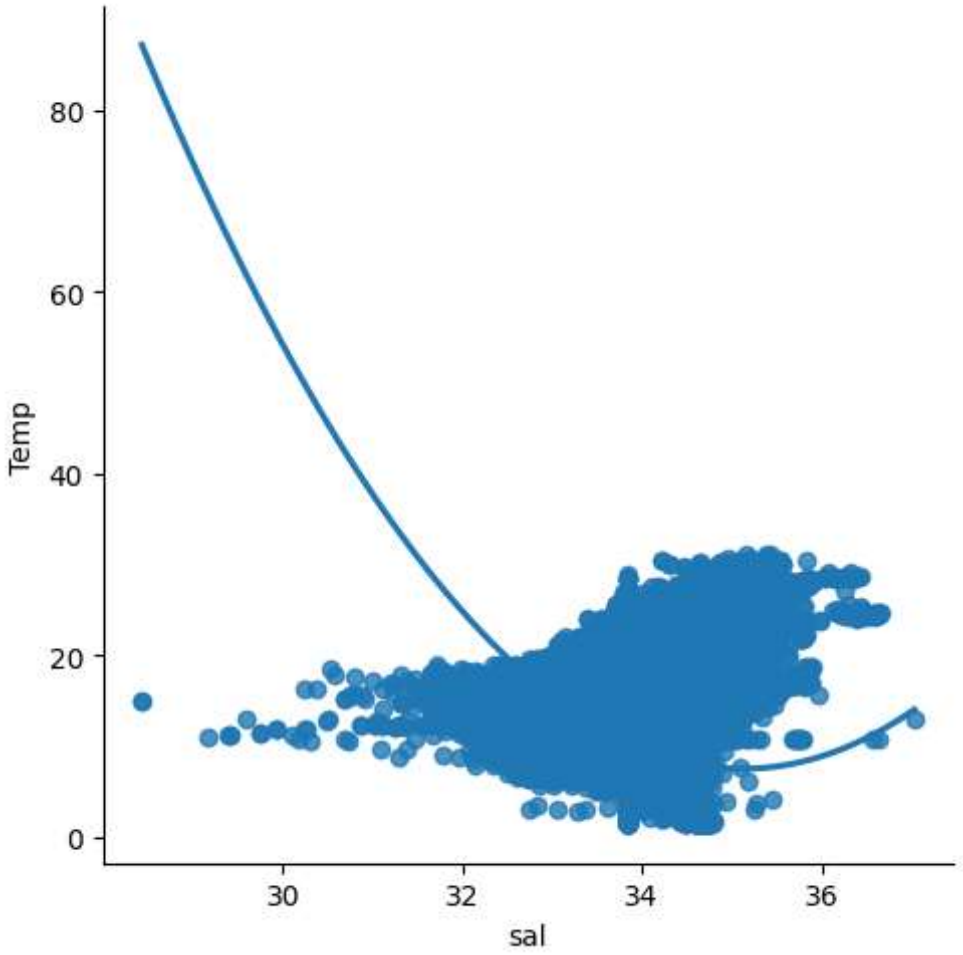
```
sal     False
Temp    False
dtype: bool
```

In [44]:

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

Out[44]:

```
<seaborn.axisgrid.FacetGrid at 0x1bd8231d890>
```



In [47]:

```
df.describe()
```

Out[47]:

	sal	Temp
count	864863.000000	864863.000000
mean	33.840350	10.799677
std	0.449022	4.216841
min	28.431000	1.440000
25%	33.504000	7.720000
50%	33.840350	10.130000
75%	34.180000	13.830000
max	37.034000	31.140000

In [48]:

```
df.fillna(method='ffill',inplace=True)
```

```
C:\Users\prajapath Arjun\AppData\Local\Temp\ipykernel_25184\4116506308.py:
1: SettingWithCopyWarning:
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.fillna(method='ffill',inplace=True)
```

In [49]:

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

In [50]:

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

```
C:\Users\prajapath Arjun\AppData\Local\Temp\ipykernel_25184\1379821321.py:
1: SettingWithCopyWarning:
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 [55]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
```

In [56]:

```
regr=LinearRegression()
regr.fit(x_train,y_train)
print(regr.score(x_test,y_test))
```

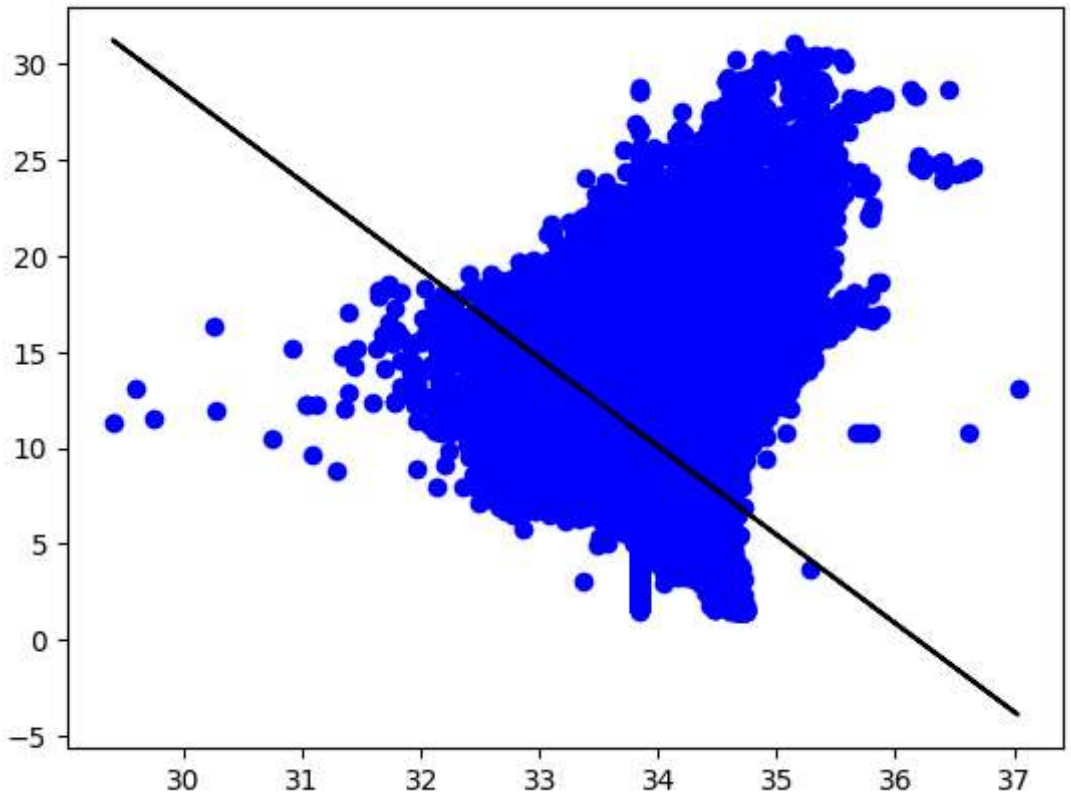
```
0.23923796511229323
```

In [57]:

```
y_pred=regr.predict(x_test)
```

In [59]:

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



In [60]:

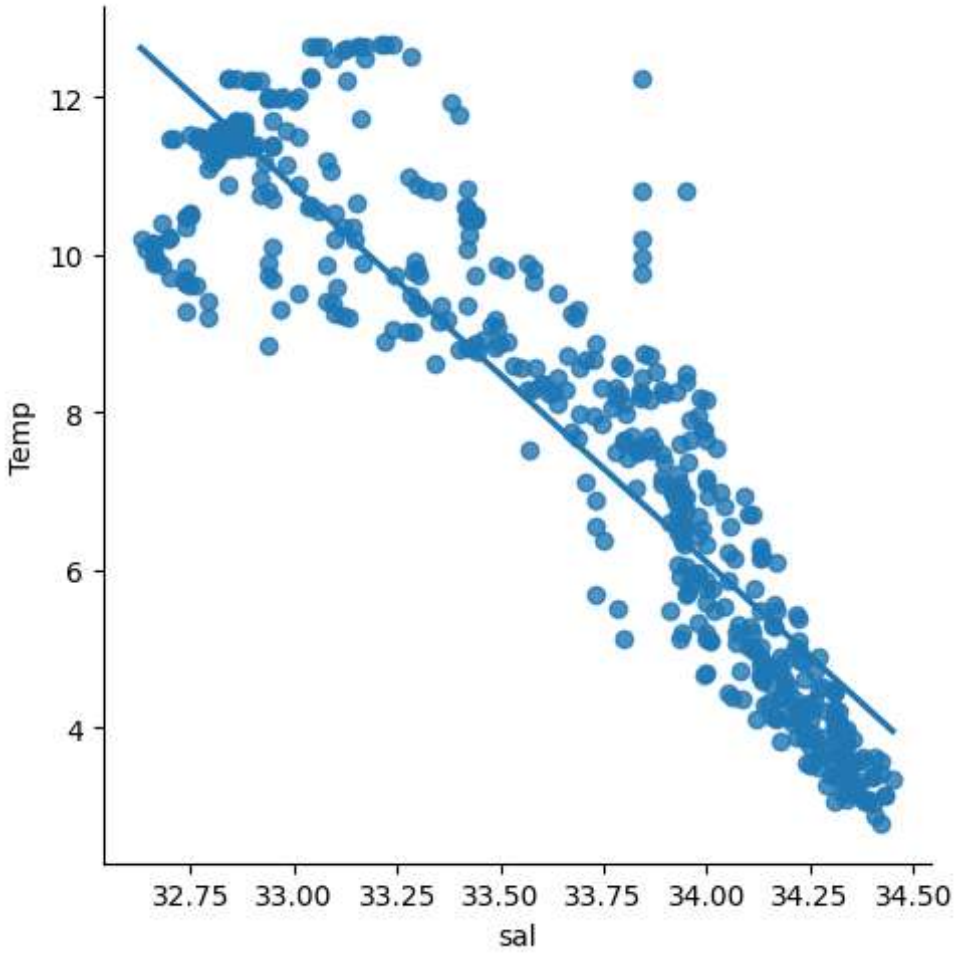
```
df500=df[:][:500]
```

In [63]:

```
sns.lmplot(x="sal",y="Temp",data=df500,order=1,ci=None)
```

Out[63]:

```
<seaborn.axisgrid.FacetGrid at 0x1bd832cbb50>
```

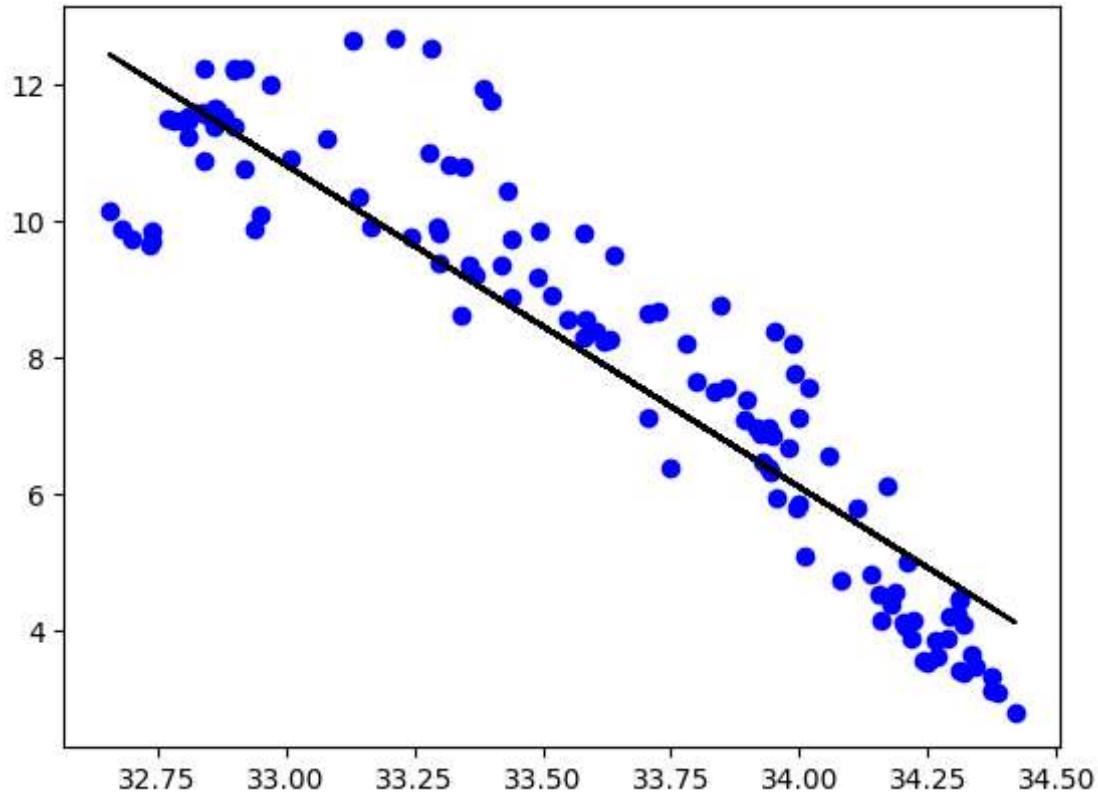



```
df500.fillna(method='ffill',inplace=True)
x=np.array(df500['sal']).reshape(-1,1)
y=np.array(df500['Temp']).reshape(-1,1)
df500.dropna(inplace=True)
```

In [70]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
regr.fit(x_train,y_train)
print("Regression:",regr.score(x_test,y_test))
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```

```
Regression: 0.8470924491957248
```



In [71]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
model=LinearRegression()
model.fit(x_train,y_train)
```

Out[71]:

```
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.