Linear Regression

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

```
C:\Users\chila\AppData\Local\Temp\ipykernel_15148\2214167869.py:8: DtypeWa
rning: Columns (47,73) have mixed types. Specify dtype option on import or
set low_memory=False.
  df=pd.read_csv(r"C:\Users\chila\Downloads\bottle.csv.zip")
```

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 864863 entries, 0 to 864862
Data columns (total 74 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   Cst_Cnt              864863 non-null  int64
 1   Btl_Cnt              864863 non-null  int64
 2   Sta_ID               864863 non-null  object
 3   Depth_ID             864863 non-null  object
 4   Depthm               864863 non-null  int64
 5   T_degC               853900 non-null  float64
 6   Salnty               817509 non-null  float64
 7   O2ml_L               696201 non-null  float64
 8   STheta               812174 non-null  float64
 9   O2Sat                661274 non-null  float64
 10  Oxy_µmol/Kg          661268 non-null  float64
 11  BtlNum               118667 non-null  float64
 12  RecInd               864863 non-null  int64
 13  T_prec               853900 non-null  float64
 14  T_qual               23127 non-null   float64
 15  S_prec               817509 non-null  float64
 16  S_qual               74914 non-null   float64
 17  P_qual               673755 non-null  float64
 18  O_qual               184676 non-null  float64
 19  SThtaq               65823 non-null   float64
 20  O2Satq               217797 non-null  float64
 21  ChlorA               225272 non-null  float64
 22  Chlqua               639166 non-null  float64
 23  Phaeop               225271 non-null  float64
 24  Phaqua               639170 non-null  float64
 25  PO4uM                413317 non-null  float64
 26  PO4q                 451786 non-null  float64
 27  SiO3uM               354091 non-null  float64
 28  SiO3qu               510866 non-null  float64
 29  NO2uM                337576 non-null  float64
 30  NO2q                 529474 non-null  float64
 31  NO3uM                337403 non-null  float64
 32  NO3q                 529933 non-null  float64
 33  NH3uM                64962 non-null   float64
 34  NH3q                 808299 non-null  float64
 35  C14As1               14432 non-null   float64
 36  C14A1p               12760 non-null   float64
 37  C14A1q               848605 non-null  float64
 38  C14As2               14414 non-null   float64
 39  C14A2p               12742 non-null   float64
 40  C14A2q               848623 non-null  float64
 41  DarkAs               22649 non-null   float64
 42  DarkAp               20457 non-null   float64
 43  DarkAq               840440 non-null  float64
 44  MeanAs               22650 non-null   float64
 45  MeanAp               20457 non-null   float64
 46  MeanAq               840439 non-null  float64
 47  IncTim               14437 non-null   object
 48  LightP               18651 non-null   float64
 49  R_Depth              864863 non-null  float64
 50  R_TEMP               853900 non-null  float64
 51  R_POTEMP             818816 non-null  float64
 52  R_SALINITY           817509 non-null  float64
 53  R_SIGMA              812007 non-null  float64
 54  R_SVA                812092 non-null  float64
 55  R_DYNHT              818206 non-null  float64
```

```
 56  R_O2                  696201 non-null  float64
 57  R_O2Sat               666448 non-null  float64
 58  R_SIO3                354099 non-null  float64
 59  R_PO4                 413325 non-null  float64
 60  R_NO3                 337411 non-null  float64
 61  R_NO2                 337584 non-null  float64
 62  R_NH4                 64982 non-null   float64
 63  R_CHLA                225276 non-null  float64
 64  R_PHAEO               225275 non-null  float64
 65  R_PRES                864863 non-null  int64
 66  R_SAMP                122006 non-null  float64
 67  DIC1                  1999 non-null    float64
 68  DIC2                  224 non-null     float64
 69  TA1                   2084 non-null    float64
 70  TA2                   234 non-null     float64
 71  pH2                   10 non-null      float64
 72  pH1                   84 non-null      float64
 73  DIC Quality Comment   55 non-null      object
dtypes: float64(65), int64(5), object(4)
memory usage: 488.3+ MB
```

In [4]:

```
df.describe()
```

Out[4]:

	Cst_Cnt	Btl_Cnt	Depthm	T_degC	Salnty	O
count	864863.000000	864863.000000	864863.000000	853900.000000	817509.000000	696201.0
mean	17138.790958	432432.000000	226.831951	10.799677	33.840350	3.3
std	10240.949817	249664.587269	316.050259	4.243825	0.461843	2.0
min	1.000000	1.000000	0.000000	1.440000	28.431000	-0.0
25%	8269.000000	216216.500000	46.000000	7.680000	33.488000	1.3
50%	16848.000000	432432.000000	125.000000	10.060000	33.863000	3.4
75%	26557.000000	648647.500000	300.000000	13.880000	34.196900	5.5
max	34404.000000	864863.000000	5351.000000	31.140000	37.034000	11.1

8 rows × 70 columns

In [5]:

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

Out[5]:

```
Cst_Cnt                False
Btl_Cnt                False
Sta_ID                 False
Depth_ID               False
Depthm                 False
                       ...
TA1                     True
TA2                     True
pH2                     True
pH1                     True
DIC Quality Comment     True
Length: 74, dtype: bool
```

In [6]:

```
df.isnull().sum()
```

Out[6]:

```
Cst_Cnt                    0
Btl_Cnt                    0
Sta_ID                     0
Depth_ID                   0
Depthm                     0
                         ...
TA1                   862779
TA2                   864629
pH2                   864853
pH1                   864779
DIC Quality Comment   864808
Length: 74, dtype: int64
```

In [7]:

```
df.head()
```

Out[7]:

	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 rows × 74 columns

In [8]:

```
df.tail()
```

Out[8]:

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

5 rows × 74 columns

In [9]:

```
df.count()
```

Out[9]:

```
Cst_Cnt                 864863
Btl_Cnt                 864863
Sta_ID                  864863
Depth_ID                864863
Depthm                  864863
                         ...
TA1                       2084
TA2                        234
pH2                         10
pH1                         84
DIC Quality Comment         55
Length: 74, dtype: int64
```

In [10]:

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

In [11]:

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

Out[11]:

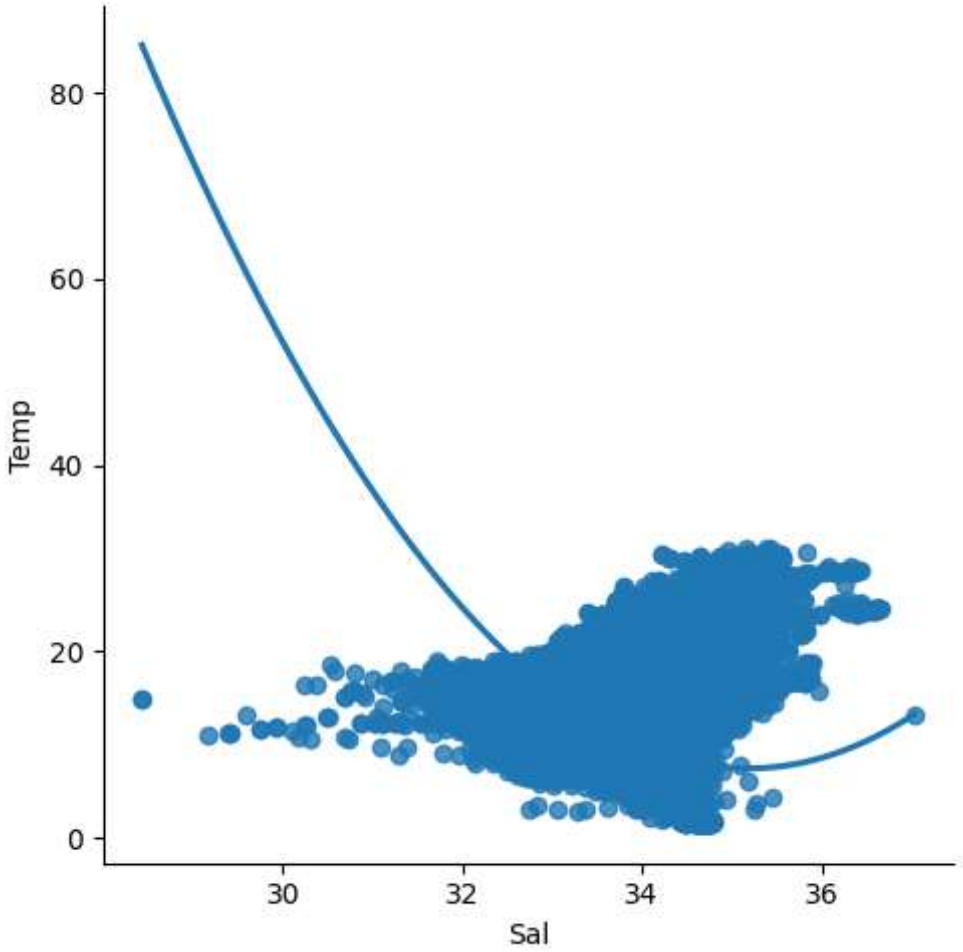
	Sal	Temp
0	33.440	10.50
1	33.440	10.46
2	33.437	10.46
3	33.420	10.45
4	33.421	10.45
5	33.431	10.45
6	33.440	10.45
7	33.424	10.24
8	33.420	10.06
9	33.494	9.86

In [12]:

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

Out[12]:

```
<seaborn.axisgrid.FacetGrid at 0x23b871ab130>
```



In [19]:

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

Out[19]:

	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]:

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

In [21]:

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

In [22]:

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

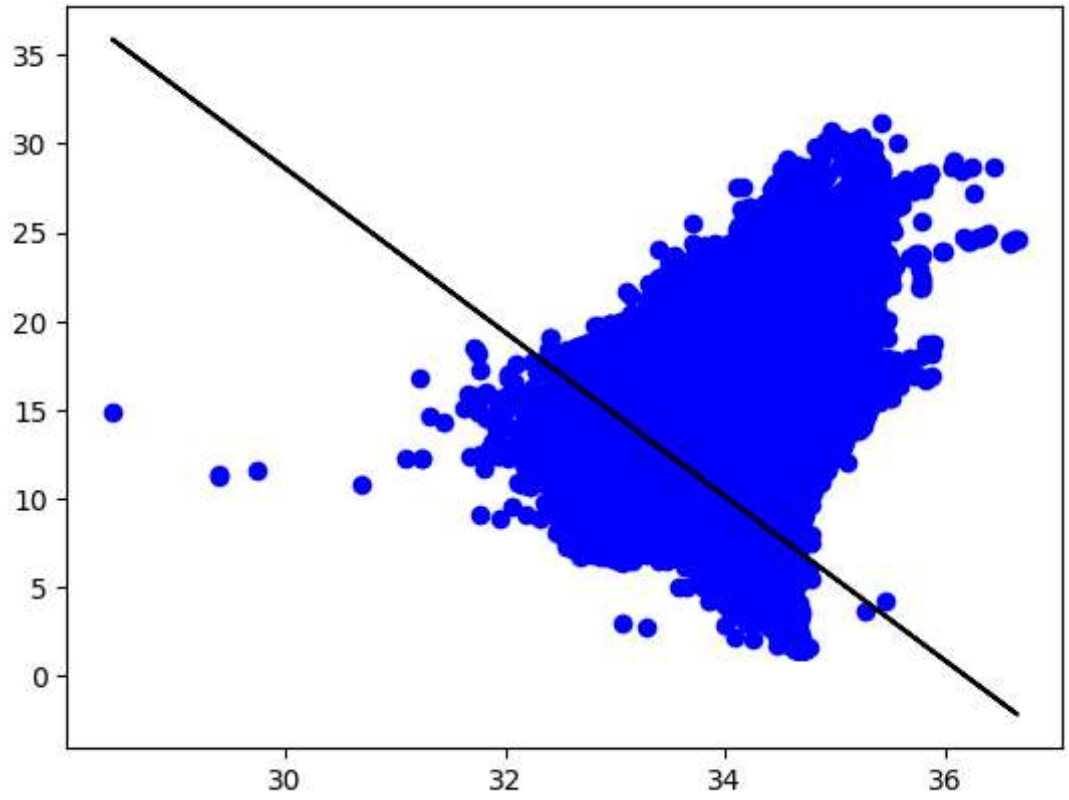
In [23]:

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

```
0.25524220994745517
```

In [24]:

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

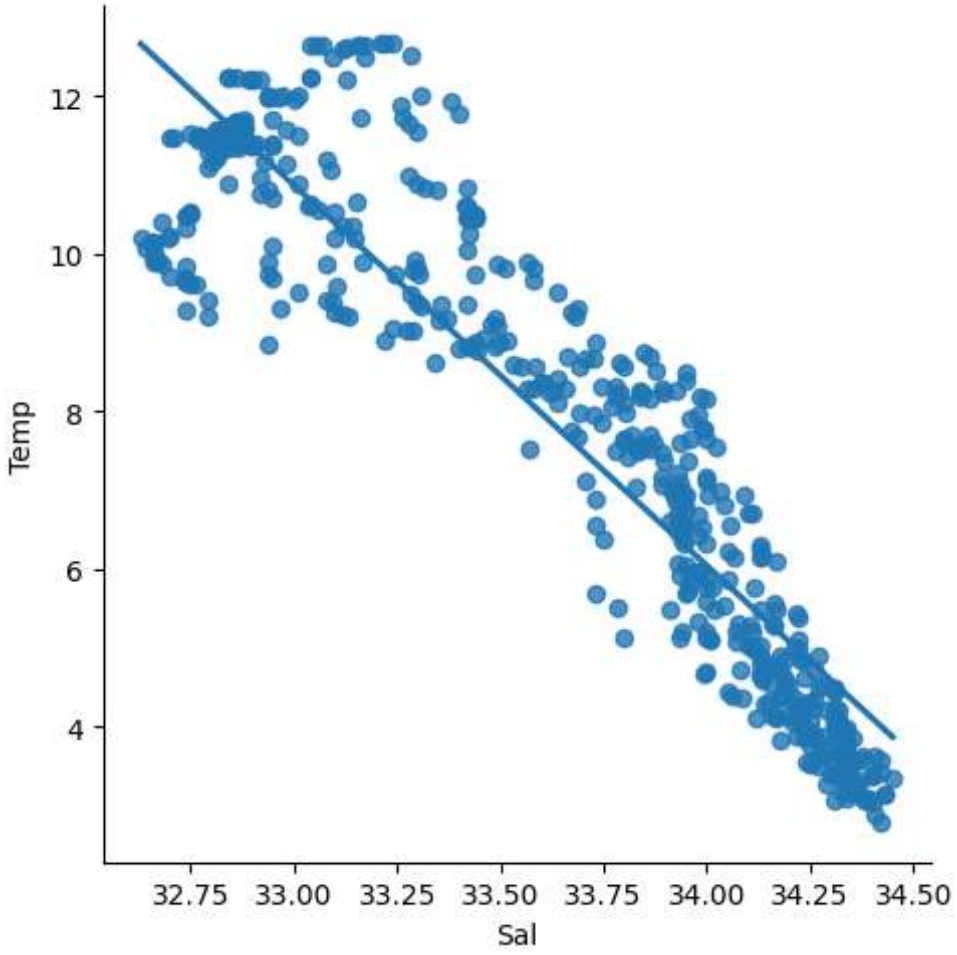


In [25]:

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

Out[25]:

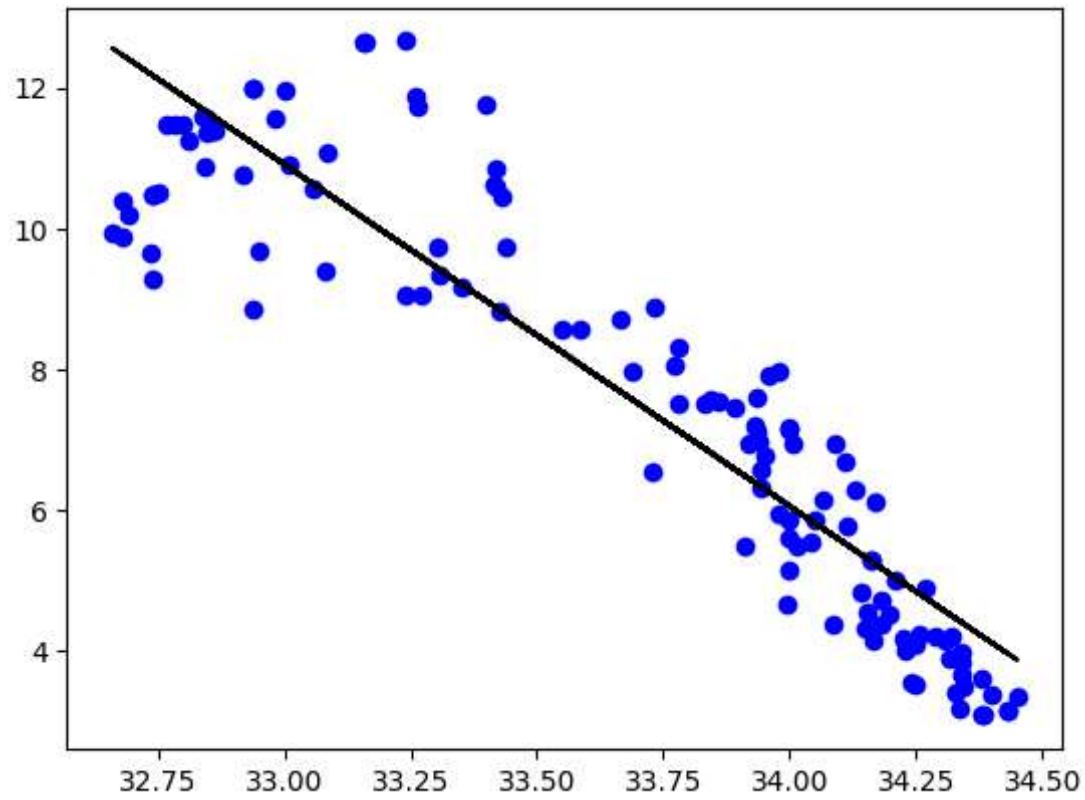
```
<seaborn.axisgrid.FacetGrid at 0x23b8209b0d0>
```



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)
regr=LinearRegression ()
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: 0.8424804357214665
```



In [27]:

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

```
R2 score: 0.8424804357214665
```

conclusion:

Linear regression is not fit for the model