linear regression model: 1)Problem statement:how best fit the dataset?

In [4]:

```
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
```

2)Reading data

In [5]:
```
df=pd.read_csv(r"C:\Users\Jayadeep\Downloads\bottle.csv.zip")
df
```

```
C:\Users\Jayadeep\AppData\Local\Temp\ipykernel_26344\2871314872.py:1: DtypeWarning: Columns (47,73) have mixed type
s. Specify dtype option on import or set low_memory=False.
  df=pd.read_csv(r"C:\Users\Jayadeep\Downloads\bottle.csv.zip")
```

Out[5]:

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	...	R_PHAEO	R_PRES	R_SAMP	DIC1	DIC2
0	1	1	054.0 056.0	19-4903CR-HY-060-0930-05400560-0000A-3	0	10.500	33.4400	NaN	25.64900	NaN	...	NaN	0	NaN	NaN	NaN
1	1	2	054.0 056.0	19-4903CR-HY-060-0930-05400560-0008A-3	8	10.460	33.4400	NaN	25.65600	NaN	...	NaN	8	NaN	NaN	NaN
2	1	3	054.0 056.0	19-4903CR-HY-060-0930-05400560-0010A-7	10	10.460	33.4370	NaN	25.65400	NaN	...	NaN	10	NaN	NaN	NaN
3	1	4	054.0 056.0	19-4903CR-HY-060-0930-05400560-0019A-3	19	10.450	33.4200	NaN	25.64300	NaN	...	NaN	19	NaN	NaN	NaN
4	1	5	054.0 056.0	19-4903CR-HY-060-0930-05400560-0020A-7	20	10.450	33.4210	NaN	25.64300	NaN	...	NaN	20	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
864858	34404	864859	093.4 026.4	20-1611SR-MX-310-2239-09340264-0000A-7	0	18.744	33.4083	5.805	23.87055	108.74	...	0.18	0	NaN	NaN	NaN

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	...	R_PHAEO	R_PRES	R_SAMP	DIC1	DIC2
864859	34404	864860	093.4 026.4	20-1611SR-MX-310-2239-09340264-0002A-3	2	18.744	33.4083	5.805	23.87072	108.74	...	0.18	2	4.0	NaN	NaN
864860	34404	864861	093.4 026.4	20-1611SR-MX-310-2239-09340264-0005A-3	5	18.692	33.4150	5.796	23.88911	108.46	...	0.18	5	3.0	NaN	NaN
864861	34404	864862	093.4 026.4	20-1611SR-MX-310-2239-09340264-0010A-3	10	18.161	33.4062	5.816	24.01426	107.74	...	0.31	10	2.0	NaN	NaN
864862	34404	864863	093.4 026.4	20-1611SR-MX-310-2239-09340264-0015A-3	15	17.533	33.3880	5.774	24.15297	105.66	...	0.61	15	1.0	NaN	NaN

864863 rows × 74 columns

In [8]:
```
df=df[['Salnty','T_degC']]
df.columns=['sal','temp']
```

In [9]:
```
df.head(15)
```

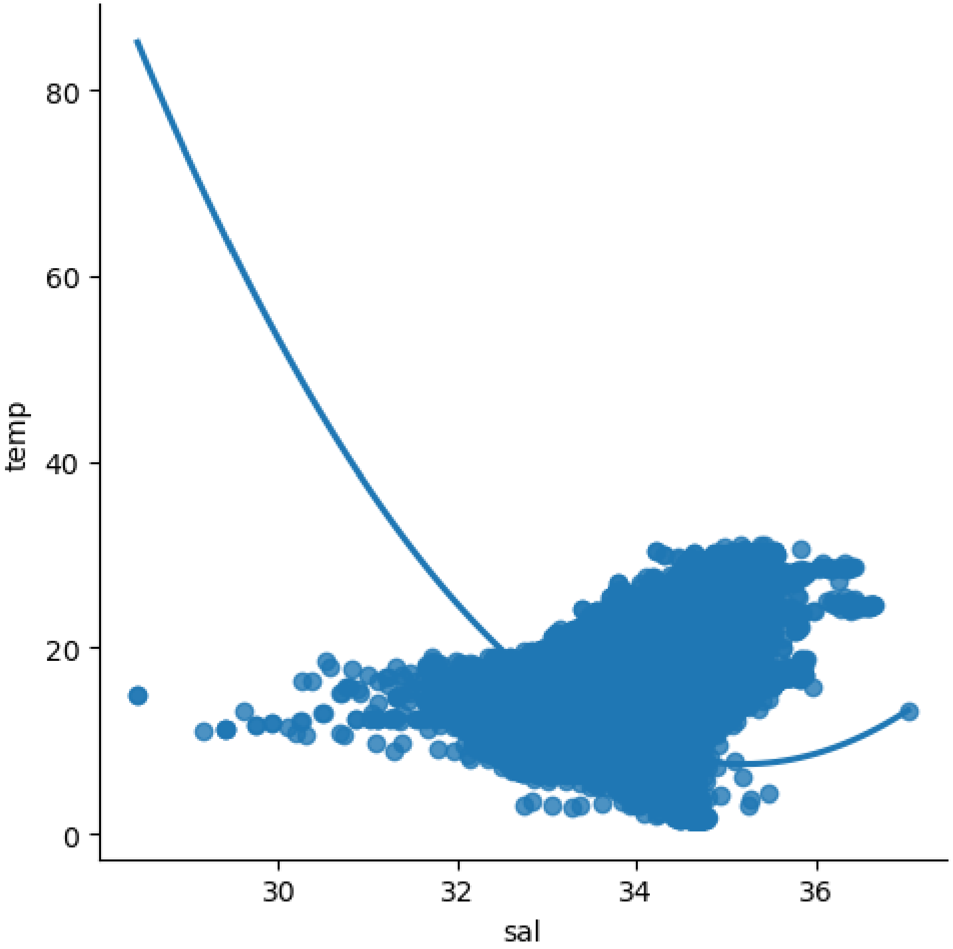
Out[9]:

	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
10	33.510	9.83
11	33.580	9.67
12	33.640	9.50
13	33.689	9.32
14	33.847	8.76

3)Exploing the data scatter-plottting the data scatter

In [11]:
```
sns.lmplot(x='sal',y='temp',data=df,order=2,ci=None)
```

Out[11]:
```
<seaborn.axisgrid.FacetGrid at 0x1e9f42e9220>
```



In [12]:
```
df.describe()
```

Out[12]:

	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 [13]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 864863 entries, 0 to 864862
Data columns (total 2 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   sal     817509 non-null  float64
 1   temp    853900 non-null  float64
dtypes: float64(2)
memory usage: 13.2 MB
```

4)Data cleaning-Eliminating nan or missing i/p numbers

In [15]:

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

```
C:\Users\Jayadeep\AppData\Local\Temp\ipykernel_26344\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-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versu
s-a-copy)
  df.fillna(method='ffill',inplace=True)
```

5)Training our model

In [16]:

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

In [17]:

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

```
C:\Users\Jayadeep\AppData\Local\Temp\ipykernel_26344\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-docs/stable/user_guide/indexing.html#returnin
g-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versu
s-a-copy)
  df.dropna(inplace=True)
```

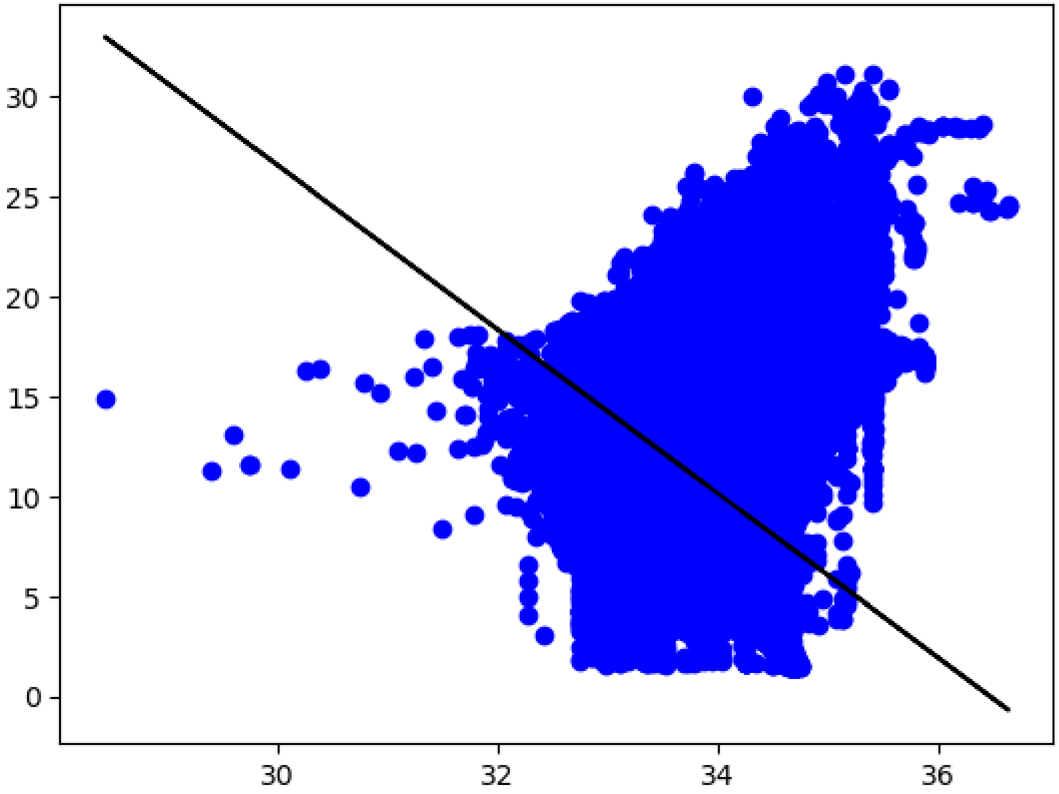
In [18]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
#splitting data into train and test
regr=LinearRegression()
regr.fit(x_train,y_train)
print(regr.score(x_test,y_test))
```

```
0.20364539273594318
```

6)Exploring our results

In [19]:
```
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```



7)working with smaller dataset

In [20]:
```
df500=df[:][:500]
sns.lmplot(x="sal",y="temp",data=df500,order=1,ci=None)
```

Out[20]:
```
<seaborn.axisgrid.FacetGrid at 0x1e9f4071d30>
```

In [21]:

```
df500.fillna(method='ffill',inplace=True)
```

In [22]:

```
x=np.array(df500['sal']).reshape(-1,1)
y=np.array(df500['temp']).reshape(-1,1)
```

In [23]:

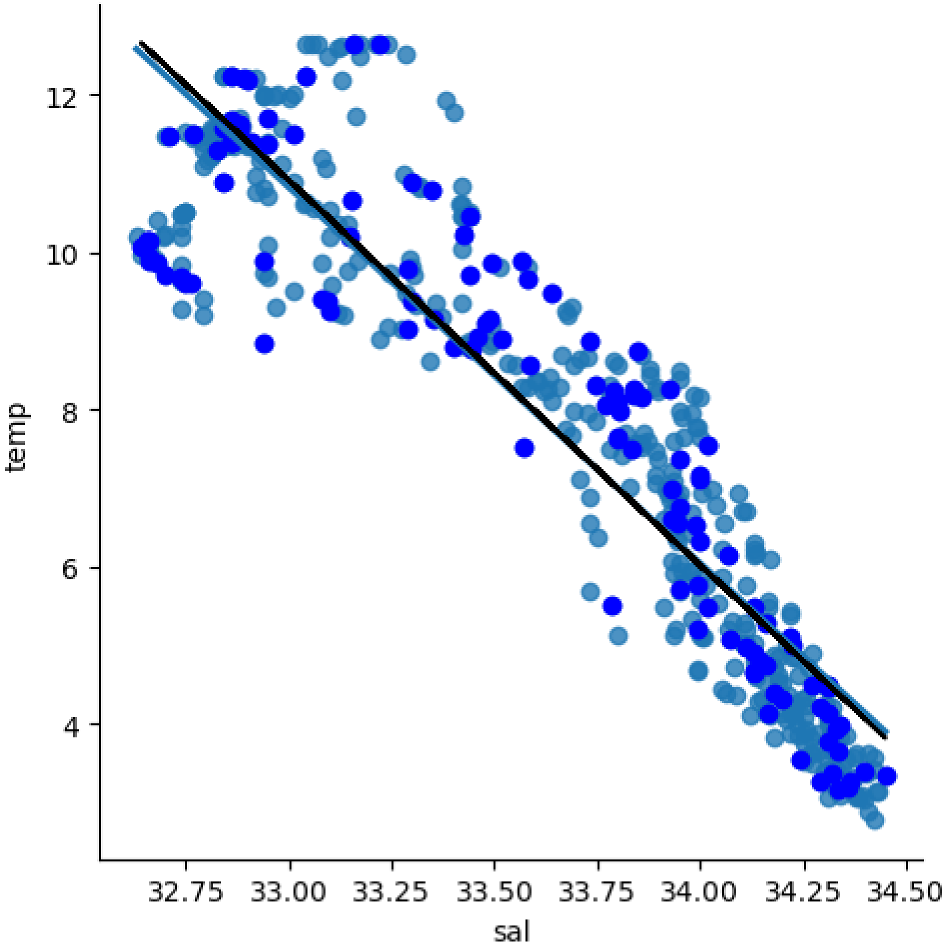
```
df500.dropna(inplace=True)
```

In [25]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
#splitting data into train and test
regr=LinearRegression()
regr.fit(x_train,y_train)
print('Regression:',regr.score(x_test,y_test))
```

```
Regression: 0.8173107904985126
```

```
In [26]: y_pred=regr.predict(x_test)
         plt.scatter(x_test,y_test,color='b')
         plt.plot(x_test,y_pred,color='k')
         plt.show()
```



8)Evaluation of model

In [27]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
```

In [28]:

```
#train model
model=LinearRegression()
model.fit(x_train,y_train)
#Evaluation the model on the test set
y_pred=model.predict(x_test)
r2=r2_score(y_test,y_pred)
print("R2 score:",r2)
```

```
R2 score: 0.8173107904985126
```

conclusion: Data set we have taken is poor for this model,but smaller data is suitable for linear model