In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import preprocessing,svm
```

In [3]:
```
df=pd.read_csv(r"C:\Users\venka\Downloads\bottle.csv\bottle.csv")
print(df)
```

```
C:\Users\venka\AppData\Local\Temp\ipykernel_10296\1869258293.py:1: DtypeWarn
ing: Columns (47,73) have mixed types. Specify dtype option on import or set
low_memory=False.
  df=pd.read_csv(r"C:\Users\venka\Downloads\bottle.csv\bottle.csv")
```

```
        Cst_Cnt  Btl_Cnt       Sta_ID                                Depth_I
D
0             1        1  054.0 056.0  19-4903CR-HY-060-0930-05400560-0000A-
3  \
1             1        2  054.0 056.0  19-4903CR-HY-060-0930-05400560-0008A-
3
2             1        3  054.0 056.0  19-4903CR-HY-060-0930-05400560-0010A-
7
3             1        4  054.0 056.0  19-4903CR-HY-060-0930-05400560-0019A-
3
4             1        5  054.0 056.0  19-4903CR-HY-060-0930-05400560-0020A-
7
...         ...      ...          ...
...
864858    34404   864859  093.4 026.4  20-1611SR-MX-310-2239-09340264-0000A-
7
864859    34404   864860  093.4 026.4  20-1611SR-MX-310-2239-09340264-0002A-
3
864860    34404   864861  093.4 026.4  20-1611SR-MX-310-2239-09340264-0005A-
3
864861    34404   864862  093.4 026.4  20-1611SR-MX-310-2239-09340264-0010A-
3
864862    34404   864863  093.4 026.4  20-1611SR-MX-310-2239-09340264-0015A-
3

        Depthm  T_degC   Salnty  O2ml_L    STheta   O2Sat  ...  R_PHAEO
0            0  10.500  33.4400     NaN  25.64900     NaN  ...      NaN  \
1            8  10.460  33.4400     NaN  25.65600     NaN  ...      NaN
2           10  10.460  33.4370     NaN  25.65400     NaN  ...      NaN
3           19  10.450  33.4200     NaN  25.64300     NaN  ...      NaN
4           20  10.450  33.4210     NaN  25.64300     NaN  ...      NaN
...        ...     ...      ...     ...       ...     ...  ...      ...
864858       0  18.744  33.4083   5.805  23.87055  108.74  ...     0.18
864859       2  18.744  33.4083   5.805  23.87072  108.74  ...     0.18
864860       5  18.692  33.4150   5.796  23.88911  108.46  ...     0.18
864861      10  18.161  33.4062   5.816  24.01426  107.74  ...     0.31
864862      15  17.533  33.3880   5.774  24.15297  105.66  ...     0.61

        R_PRES  R_SAMP  DIC1  DIC2  TA1  TA2  pH2  pH1  DIC Quality Comment
0            0     NaN   NaN   NaN  NaN  NaN  NaN  NaN                  NaN
1            8     NaN   NaN   NaN  NaN  NaN  NaN  NaN                  NaN
2           10     NaN   NaN   NaN  NaN  NaN  NaN  NaN                  NaN
3           19     NaN   NaN   NaN  NaN  NaN  NaN  NaN                  NaN
4           20     NaN   NaN   NaN  NaN  NaN  NaN  NaN                  NaN
...        ...     ...   ...   ...  ...  ...  ...  ...                  ...
864858       0     NaN   NaN   NaN  NaN  NaN  NaN  NaN                  NaN
864859       2     4.0   NaN   NaN  NaN  NaN  NaN  NaN                  NaN
864860       5     3.0   NaN   NaN  NaN  NaN  NaN  NaN                  NaN
864861      10     2.0   NaN   NaN  NaN  NaN  NaN  NaN                  NaN
864862      15     1.0   NaN   NaN  NaN  NaN  NaN  NaN                  NaN

[864863 rows x 74 columns]
```

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

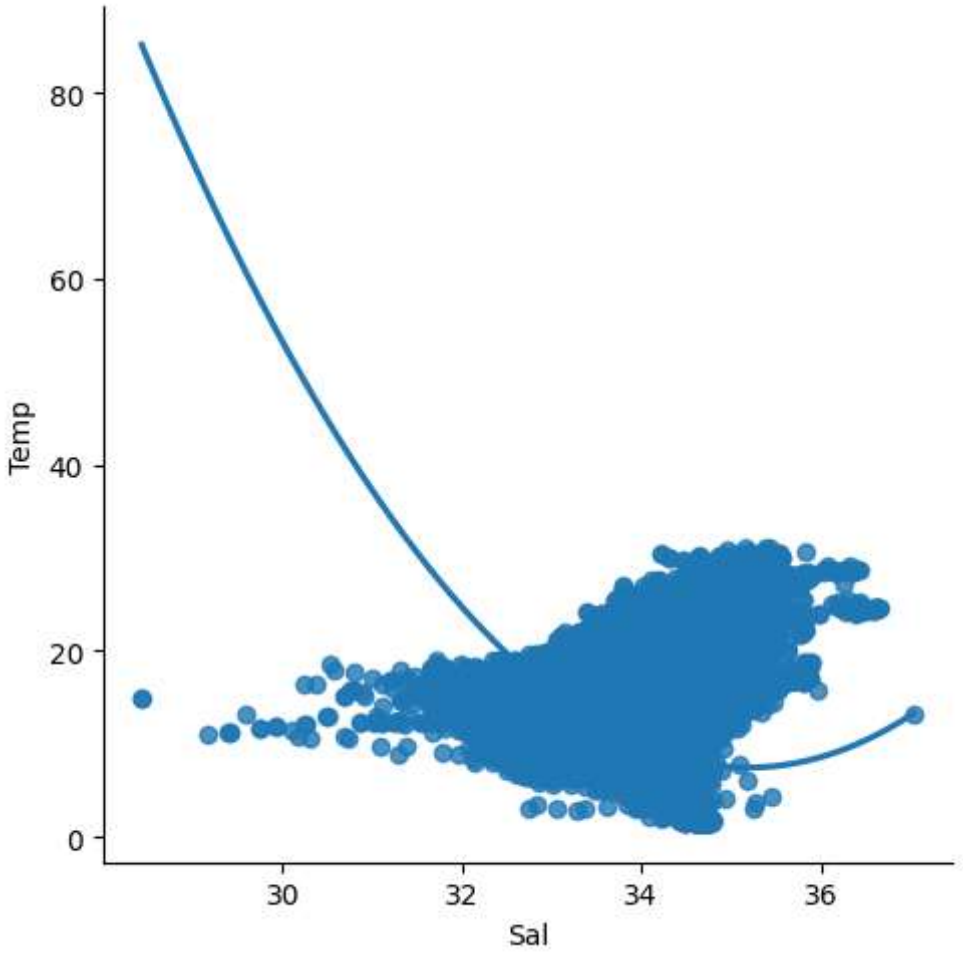
In [6]:
```
df.tail()
```

Out[6]:

	Sal	Temp
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

In [7]:
```
sns.lmplot(x="Sal",y="Temp",data=df,order=2,ci=None)
```

Out[7]:
```
<seaborn.axisgrid.FacetGrid at 0x1fef4502cd0>
```



In [8]:
```
df.describe()
```

Out[8]:

	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

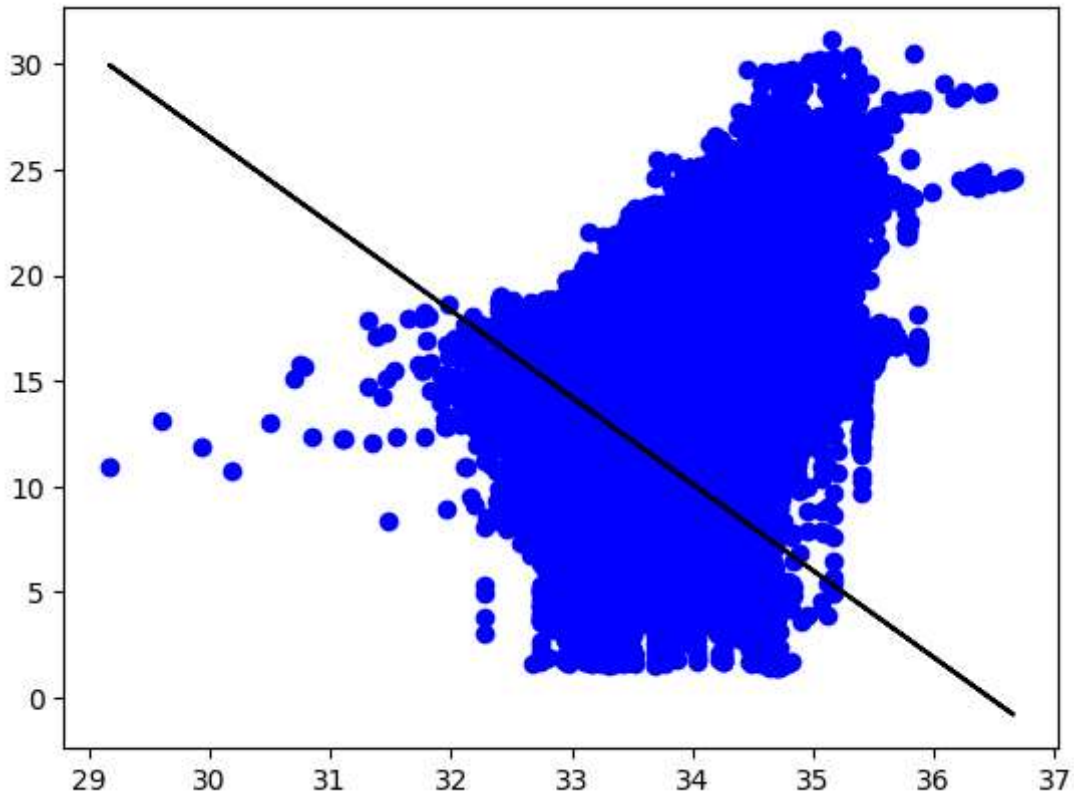
In [9]:
```
df.info()
```

```
<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
```

In [13]:
```
df.fillna(method='ffill',inplace=True)
```

In [11]:
```
df.fillna(method='ffill',inplace=True)
x=np.array(df['Sal']).reshape(-1,1)
y=np.array(df['Temp']).reshape(-1,1)
df.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:  0.20332169758048613
```

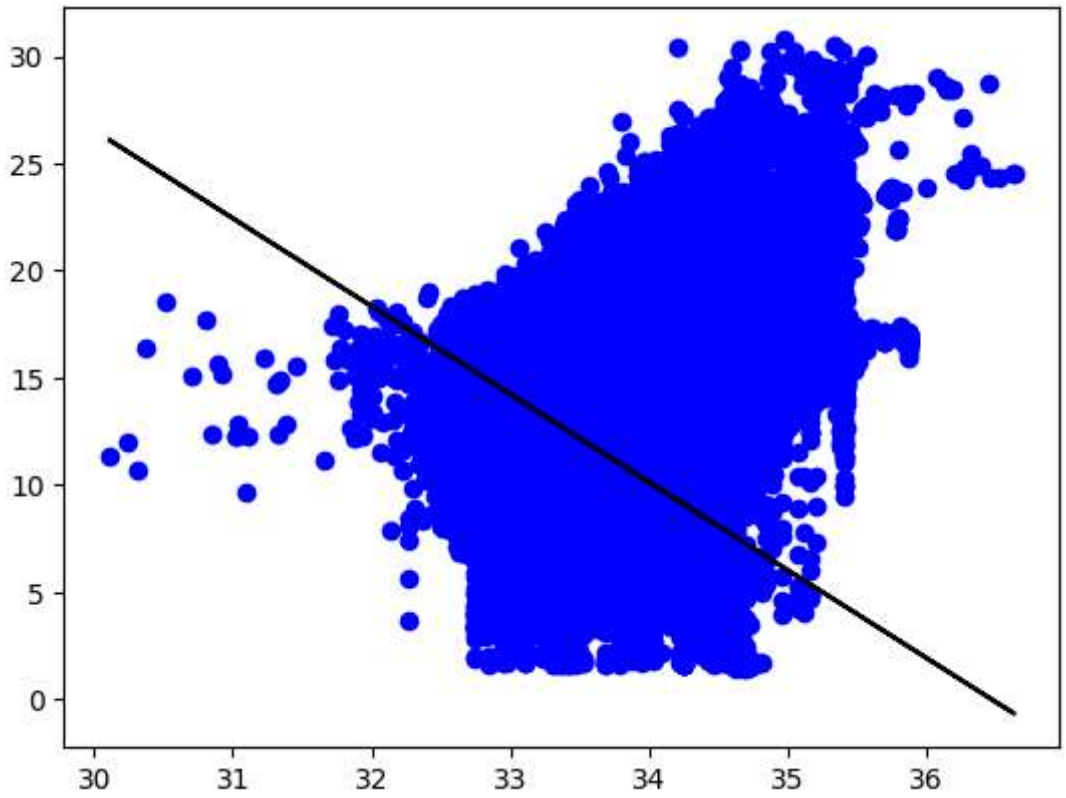


In [16]:
```
df.dropna(inplace=True)
```

In [17]:
```
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(regr.score(x_test,y_test))
```

```
0.20276461340567697
```

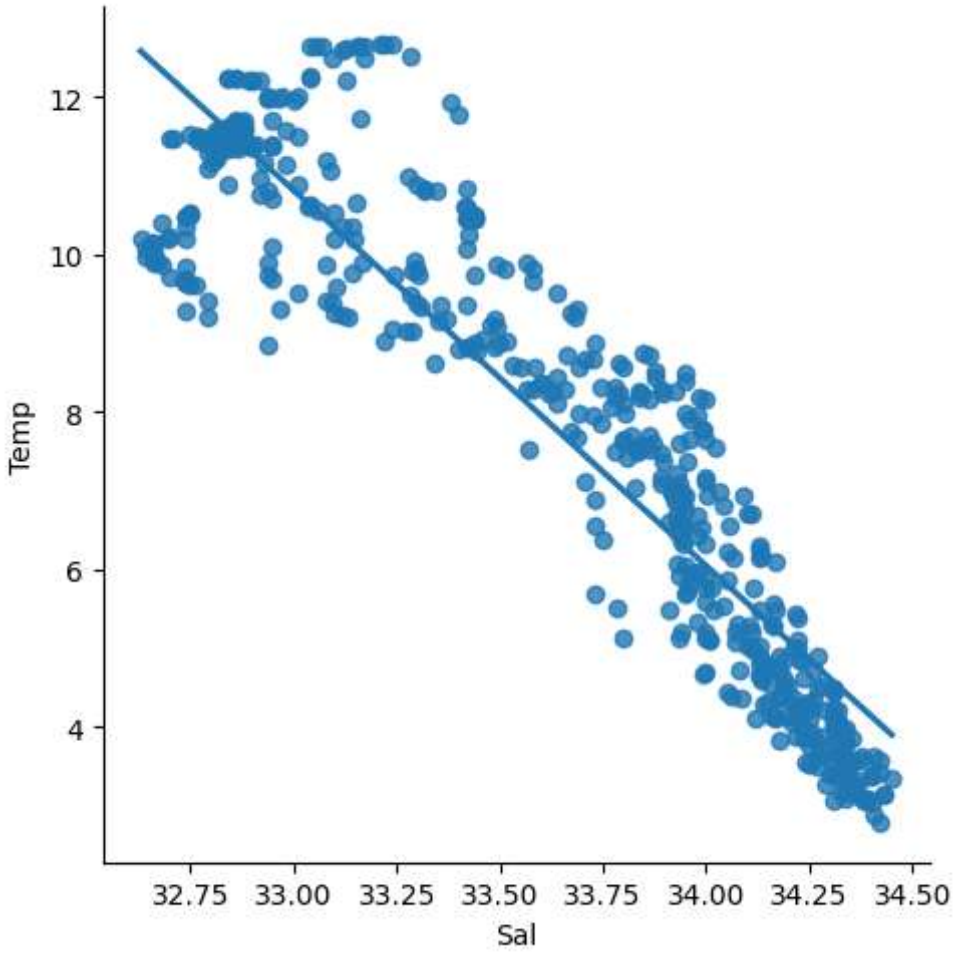
In [18]:
```
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```



In [19]:
```
df=df[:][:500]
```

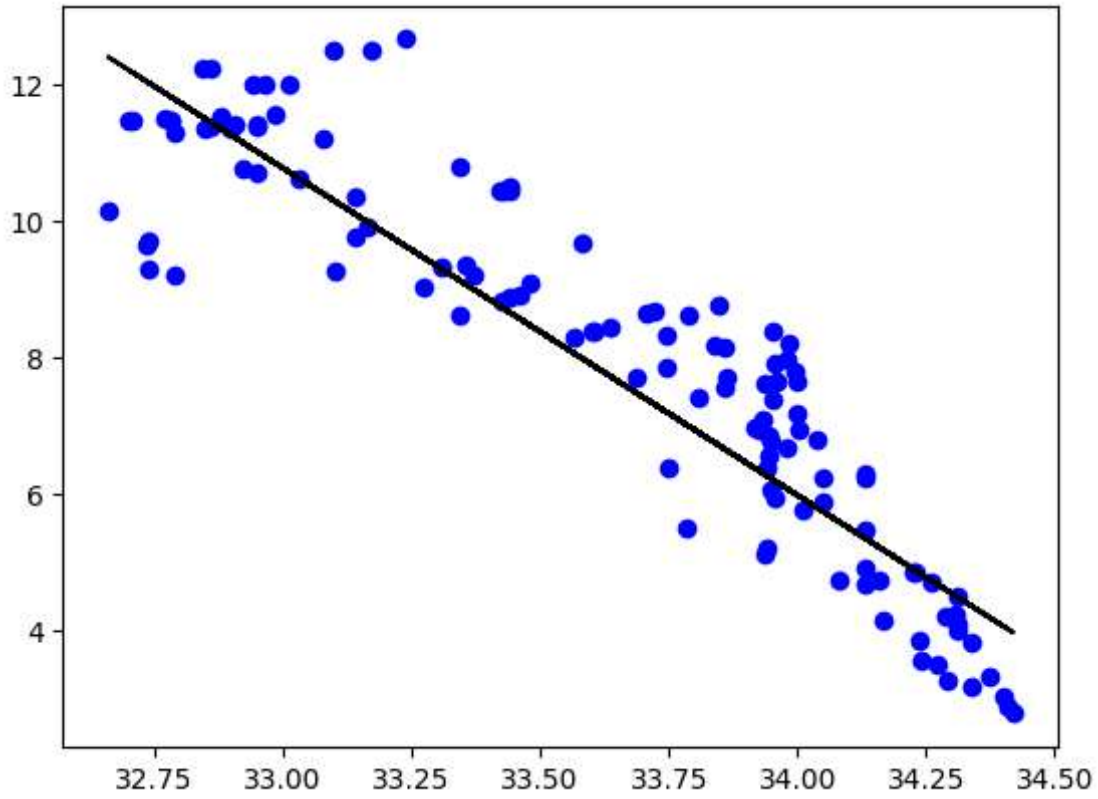
In [20]:
```
sns.lmplot(x="Sal",y="Temp",data=df,order=1,ci=None)
```

Out[20]: <seaborn.axisgrid.FacetGrid at 0x1fef8096090>



In [21]:
```
df.fillna(method='ffill',inplace=True)
x=np.array(df['Sal']).reshape(-1,1)
y=np.array(df['Temp']).reshape(-1,1)
df.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:  0.8276840473714211
```



In [22]:
```
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: 0.8276840473714211
```

In [ ]:
```

```