

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
```


	33.440	10.50								
1	33.440	10.46			20-1611SR-					
2	33.437	10.46	864859	093.4	MX-310-	0	18.744	33.4083	5.805	23.87055
3	33.420	10.45		026.4	2239-					
4	33.421	10.45			09340264-					
5	33.431	10.45			0000A-7					
6	33.440	10.45	864860	093.4	20-1611SR-	2	18.744	33.4083	5.805	23.87072
7	33.424	10.24		026.4	MX-310-					
8	33.420	10.06			2239-					
9	33.494	9.86			09340264-					
864860	34404	864861	093.4	MX-310-	5	18.692	33.4150	5.796	23.88911	
			026.4	2239-						
					09340264-					
					0005A-3					

In [18]:

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

```
Out[18]:
```

	34404	864862	093.4	MX-310-	10	18.161	33.4062	5.816	24.01426
sal	True		026.4	2239-					
Temp	True			09340264-					
dtype:	bool			0010A-3					

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

```
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-docs/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 [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 [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)
```

```
d+500=d+[ : ] [ : 500 ]
```


32.75 33.00 33.25 33.50 33.75 34.00 34.25 34.50
sal

```
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)
```

32.75 33.00 33.25 33.50 33.75 34.00 34.25 34.50

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.