EX:4(a) Predicting Water Temperature based on Salinity using Regression

AIM: To predict water temperature based on Salinity using Regression .

Algorithm:

1. Import the Libraries, dataset and load the dataset.
2. Read the Dataset
3. Remove the unwanted data from the dataset
4. Splitting the data into training and testing data
5. Dropping any rows with Nan values.
6. Data scatter of predicted values Exploring our results

Program:

```
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
```

Step 2: Reading the Dataset

```
df = pd.read_csv('bottle.csv')

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

# Taking only the selected two attributes from the dataset

df_binary.columns = ['Sal', 'Temp']

# Renaming the columns for easier writing of the code

df_binary.head()

# Displaying only the 1st  rows along with the column names
```

Step 3: Data Cleaning

```
# Eliminating NaN or missing input numbers

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

Step 4: Training Our Model

```
X = np.array(df_binary['Sal']).reshape(-1, 1)

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

Separating the data into independent and dependent variables

Converting each dataframe into a numpy array

since each dataframe contains only one column

df_binary.dropna(inplace = True)

Dropping any rows with Nan values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)

Splitting the data into training and testing data

regr = LinearRegression()

regr.fit(X_train, y_train)

print(regr.score(X_test, y_test))

print(df)

Step 5: Exploring Our Results

y_pred = regr.predict(X_test)

plt.scatter(X_test, y_test, color ='b')

```
plt.plot(X_test, y_pred, color ='k')

plt.show()

# Data scatter of predicted values
```

Result:

Thus the Water Temperature can be predicted on Salinity using Regression and the result is succesfully executed.