First lab of Introduction to Machine Learning

Members:

- Angel Mora
- Luis Méndez

Lab topic: Linear Regression

What is your dataset?

CalCOFI Over 60 years of oceanographic data

What is your regression problem?

Predict the water temperature based on salinity.

How many data points are there in the dataset?

There are up to 800 000 data points, we will perform a unique sampling to handle only 10 000 data points in the analysis.

```
import pandas as pd

original_dataset = pd.read_csv('dataset/bottle.csv')

original_dataset.dropna(subset=['T_degC','Salnty'], inplace=True)

sampled_dataset = original_dataset.sample(n=10000, random_state=42)

sampled_dataset.to_csv('dataset/sampled_bottle.csv', index=False)
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_28940\963850515.py:3:
DtypeWarning: Columns (47,73) have mixed types. Specify dtype option
on import or set low_memory=False.
  original_dataset = pd.read_csv('dataset/bottle.csv')
```

What is the β term assuming a zero-th point intersection (no bias)

```
import pandas as pd
from sklearn.linear_model import LinearRegression

dataset = pd.read_csv('dataset/sampled_bottle.csv', dtype={'Salnty':
float, 'T_degC': float}, low_memory=False)
selected_columns = dataset[['Salnty', 'T_degC']]

X = selected_columns[['Salnty']]
y = selected_columns[['T_degC']]
```

```
reg_nobias = LinearRegression(fit_intercept=False).fit(X, y)

β_nobias = reg_nobias.coef_[0][0]
print(f"β term with no bias: {β_nobias:.4f}")

β term with no bias: 0.3208
```

What is the bias term not assuming a zero-th point intersection (bias included)

```
reg_withbias = LinearRegression().fit(X, y)

β_withbias = reg_withbias.coef_[0][0]
bias = reg_withbias.intercept_[0]

print(f'β term with bias: {β_withbias:.4f}')
print(f'Bias: {bias:.4f}')

β term with bias: -4.7462
Bias: 171.4538
```

What are the number of independent variables?

The only independent variable is water salinity.

What is the dependent variable?

The dependent variable is water temperature.

What are the unit of measurements for each variable(s)?

The water salinity is measured in parts per thousand (ppt) and the water temperature is measured in Celsius degrees (°C).

Please plot the raw data and a superimposing line on the data that passes through the origin. What is the mean square error (MSE)?

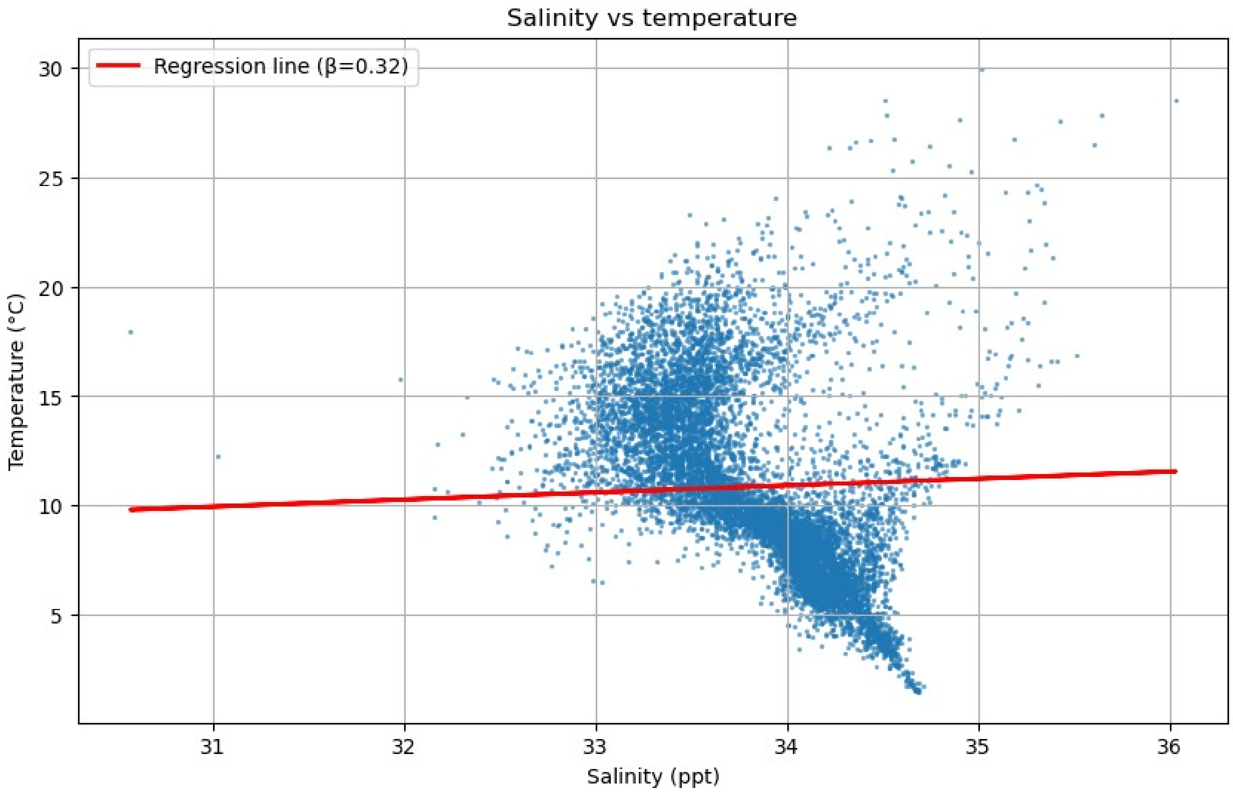
```
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error

def plot(X, y, regression, β_value):
    plt.figure(figsize=(10, 6))
    plt.scatter(X, y, s=2,alpha=0.5)
    plt.plot(X, regression, color='red', linewidth=2,
label=f'Regression line (β={β_value:.2f})')
    plt.title('Salinity vs temperature')
    plt.xlabel('Salinity (ppt)')
    plt.ylabel('Temperature (°C)')
    plt.legend()
    plt.grid(True)

predicted_y = reg_nobias.predict(X)
plot(X, y, predicted_y, β_nobias)

MSE = mean_squared_error(y, predicted_y)
print(f"Mean Square error: {MSE:.4f}")
```

```
Mean Square error: 18.0235
```

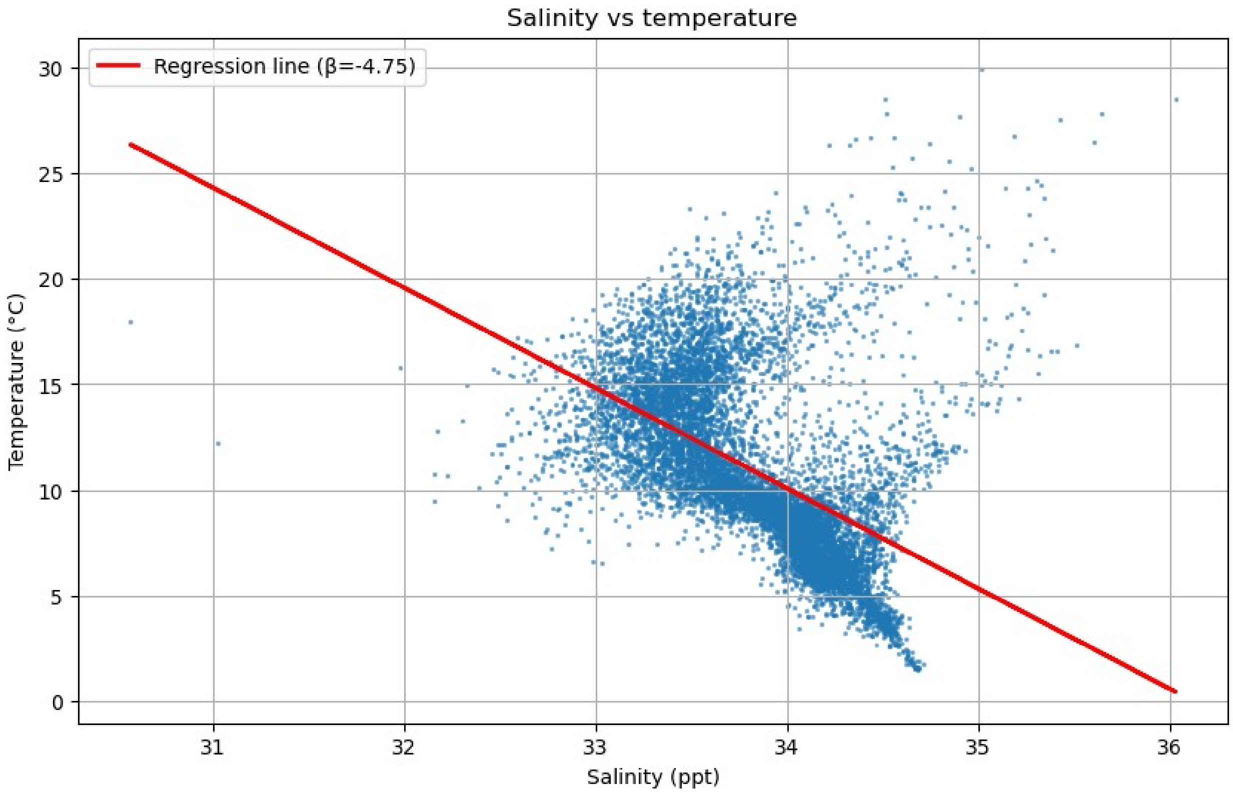


Please plot the raw data and a superimposing line on the data that does not pass through the origin. What is the mean square error (MSE)?

```
predicted_y = reg_withbias.predict(X)
plot(X, y, predicted_y, β_withbias)

MSE = mean_squared_error(y, predicted_y)
print(f"Mean Square error: {MSE:.4f}")
```

```
Mean Square error: 12.6714
```



Repeat the regression line randomly 100 times. At each iteration randomly remove with replacement one data point and perform a regression. What are the average Beta values found for both cases where the fitting is done by fitting it through the origin (bias removed), and not fitting it through the origin (bias included). Hint : Look-up statistical bootstrapping.

```
import numpy as np

lista_sin_sesgo = []
lista_con_sesgo = []

for _ in range(100):
    indices = np.random.choice(range(len(X)), len(X), replace=True)
    X_bootstrap = X.iloc[indices]
    y_bootstrap = y.iloc[indices]

lista_sin_sesgo.append(LinearRegression(fit_intercept=False).fit(X_boo
tstrap, y_bootstrap).coef_[0][0])
    lista_con_sesgo.append(LinearRegression().fit(X_bootstrap,
y_bootstrap).coef_[0][0])

print(f'Valor promedio de Beta sin sesgo:
{np.mean(lista_sin_sesgo):.4f}')
```

```
print(f'Valor promedio de Beta con sesgo:
{np.mean(lista_con_sesgo):.4f}')
```

```
Valor promedio de Beta sin sesgo: 0.3208
Valor promedio de Beta con sesgo: -4.7644
```

Please list in your 2 page report :

- **The contributions of each author.**
  - **Angel Mora**: Performed the Linear regression analysis, found MSEs and the final experiment.
  - **Luis Méndez**: Found the dataset, identify the problem, variables, data visualization (plotting) and verify results.
- **The list of all the python packages used.**
  - **Pandas**
  - **Sklearn**
  - **Matplotlib**
- **The list of all toolboxes used (and links to datasets and dataset license)**
  - CalCOFI
- **The list of any AI tools (e.g. ChatGPT, Perplexity, You) used in your homework and how.**
  - we used ChatGPT to obtain concepts about β-term and linear regression.
- **The list of all academic references used in your homework.**
  - Müller, A. C., & Guido, S. (2016). Introduction to machine learning with Python: a guide for data scientists. " O'Reilly Media, Inc.".