Multiple Linear Regression

Importing the libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

Importing the dataset

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

```
dataset = pd.read_csv('/content/drive/MyDrive/Datasets/temperat
print(dataset)
```

```
     YEAR    JAN    FEB    MAR  ...  JAN-FEB  MAR-MAY  JUN-SEP  OCT-DEC
0    1901  22.40  24.14  29.07  ...    23.27    31.46    31.27    27.25
1    1902  24.93  26.58  29.77  ...    25.75    31.76    31.09    26.49
2    1903  23.44  25.03  27.83  ...    24.24    30.71    30.92    26.26
3    1904  22.50  24.73  28.21  ...    23.62    30.95    30.66    26.40
4    1905  22.00  22.83  26.68  ...    22.25    30.00    31.33    26.57
..    ...    ...    ...    ...  ...      ...      ...      ...      ...
112  2013  24.56  26.59  30.62  ...    25.58    32.58    31.33    27.83
113  2014  23.83  25.97  28.95  ...    24.90    31.82    32.00    27.81
114  2015  24.58  26.89  29.07  ...    25.74    31.68    31.87    28.27
115  2016  26.94  29.72  32.62  ...    28.33    34.57    32.28    30.03
116  2017  26.45  29.46  31.60  ...    27.95    34.13    32.41    29.69

[117 rows x 18 columns]
```

```
dataset.head()
```

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DE
0	1901	22.40	24.14	29.07	31.91	33.41	33.18	31.21	30.39	30.47	29.97	27.31	24.4

```
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

3	1904	22.50	24.73	28.21	32.02	32.64	32.07	30.36	30.09	30.04	29.20	26.36	23.6

```
dataset.dtypes
```

```
YEAR         int64
JAN        float64
FEB        float64
MAR        float64
APR        float64
MAY        float64
JUN        float64
JUL        float64
AUG        float64
SEP        float64
OCT        float64
NOV        float64
DEC        float64
ANNUAL     float64
JAN-FEB    float64
MAR-MAY    float64
JUN-SEP    float64
OCT-DEC    float64
dtype: object
```

```
dataset.columns
```

```
Index(['YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP',
       'OCT', 'NOV', 'DEC', 'ANNUAL', 'JAN-FEB', 'MAR-MAY', 'JUN-SEP',
       'OCT-DEC'],
      dtype='object')
```

```
dataset.describe()
```

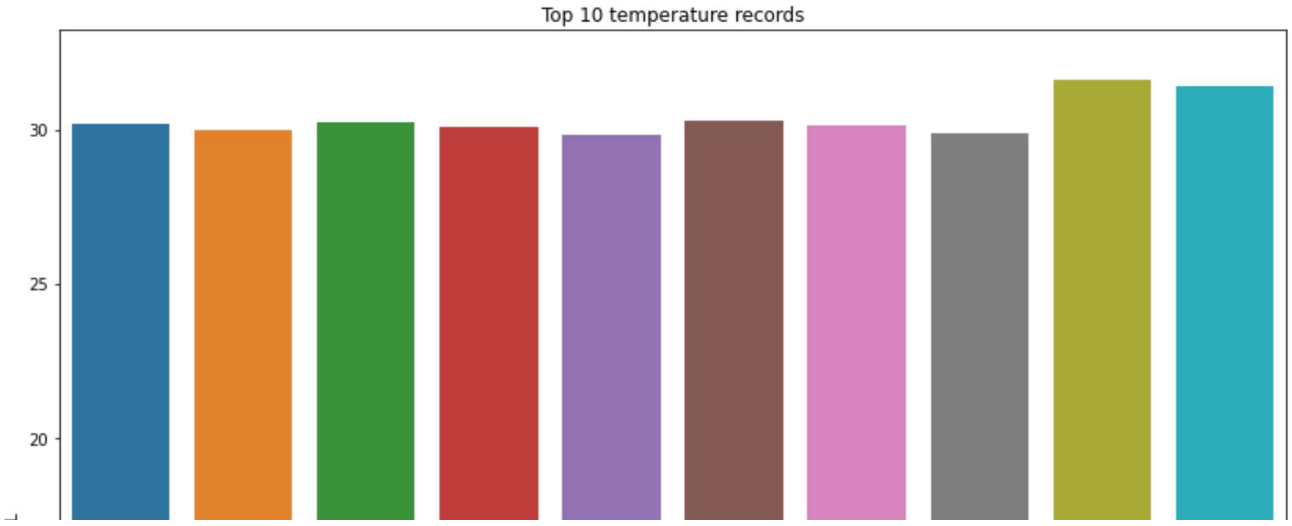
YEAR	JAN	FEB	MAR	APR	MAY	J

```
dataset.isnull().sum()
```

```
YEAR       0
JAN        0
FEB        0
MAR        0
APR        0
MAY        0
JUN        0
JUL        0
AUG        0
SEP        0
OCT        0
NOV        0
DEC        0
ANNUAL     0
JAN-FEB    0
MAR-MAY    0
JUN-SEP    0
OCT-DEC    0
dtype: int64
```

```
top_10_data = dataset.nlargest(10, "ANNUAL")
plt.figure(figsize=(14,12))
plt.title("Top 10 temperature records")
sns.barplot(x=top_10_data.YEAR, y=top_10_data.ANNUAL)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4b82d90d50>
```




```
X = dataset.iloc[:,1:13].values
y = dataset.iloc[:,13].values
```

```
print(X)
```

```
[[22.4  24.14 29.07 ... 29.97 27.31 24.49]
 [24.93 26.58 29.77 ... 29.12 26.31 24.04]
 [23.44 25.03 27.83 ... 29.04 26.08 23.65]
 ...
 [24.58 26.89 29.07 ... 31.04 28.1  25.67]
 [26.94 29.72 32.62 ... 31.98 30.11 28.01]
 [26.45 29.46 31.6  ... 32.29 29.6  27.18]]
```


```
print(y)
```

```
[28.96 29.22 28.47 28.49 28.3  28.73 28.65 28.83 28.38 28.53 28.62 28.95
 28.67 28.66 28.94 28.82 28.11 28.66 28.66 28.76 28.86 28.8  28.74 28.8
 28.67 28.7  28.59 28.98 28.76 28.65 29.15 29.09 28.49 29.03 28.76 28.71
 28.7  28.7  28.85 28.88 29.46 28.98 28.8  28.89 28.97 29.37 28.84 28.73
 28.89 28.47 29.09 29.16 29.43 28.92 28.76 28.63 28.64 29.34 29.02 29.31
 28.72 28.89 29.04 29.09 29.16 29.41 29.14 29.07 29.61 29.47 29.15 29.31
 29.44 29.26 28.89 29.27 29.41 29.23 29.63 29.58 29.32 29.12 29.11 29.28
 29.61 29.33 29.72 29.55 29.18 29.14 29.32 29.23 29.55 29.46 30.18 29.58
 29.05 29.7  29.81 29.75 29.99 30.23 29.75 29.79 29.6  30.06 29.84 29.64
 30.3  30.13 29.82 29.81 29.81 29.72 29.9  31.63 31.42]
```

Splitting the dataset into the Training set and Test set

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_
```

Training the Linear Regression model on the Training set

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Predicting the Test set results

```
y_pred = regressor.predict(X_test)
# np.set_printoptions(precision=2)
# print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.r

# print(y_test.reshape())

y_test = np.array(y_test)
y_pred = np.array(y_pred)

print(y_test)
```

```
[28.62 29.31 29.58 29.23 28.83 29.72 28.59 29.81 28.74 30.18 30.23 28.47
 29.09 28.67 31.42 29.04 29.46 28.89 28.89 29.26 28.11 30.3  28.66 28.89]
```

```
print(y_pred)
```

```
[28.61298268 29.35069218 29.67648711 29.21980812 28.80814616 29.74217943
 28.616065   29.79996582 28.71994749 30.18927669 30.26622397 28.62326581
 29.11904085 28.70726098 31.4525398  29.04149196 29.43159102 28.88563393
 28.84233802 29.23703529 28.11811715 30.34928586 28.63771382 28.86496196]
```

```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test, y_pred)
print(mse)
```

```
0.002028984063429594
```

```
import math
rmse = math.sqrt(mse)
print(rmse)
```

```
0.045044245619497214
```