Ex No.8: *Implementation of learning algorithms for an application*

Machine Learning Algorithms

Machine learning (ML) is a type of artificial intelligence that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions. There are four basic approaches : supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. The type of algorithm data scientists choose to use depends on what type of data they want to predict.

Supervised learning: In this type of machine learning, data scientists supply algorithms with labelled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting.

Steps to implement linear regression model :

1. Import required libraries

2. Define the dataset

3. Plot the data points

4. Initialize the parameters.

5. Predict the value of a dependent variable by given an independent variable.

6. Calculate the error in prediction for all data points.

7. Calculate partial derivative w.r.t a0 and a1.

8. Calculate the cost for each number and add them.

9. Update the values of a0 and a1.

Output:

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
%matplotlib inline
```

In [10]:
```
from google.colab import files
uploaded = files.upload()
import io
dataset = pd.read_csv(io.BytesIO(uploaded['student_scores.csv']))
```

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving student_scores.csv to student_scores (1).csv
```

In [11]:
```
dataset.shape
```

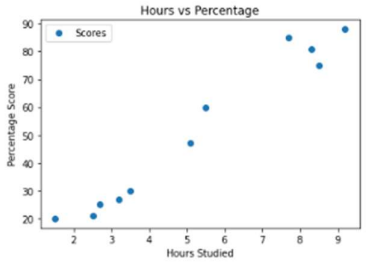
Out[11]:
```
(11, 2)
```

In [12]:
```
dataset.describe()
```

Out[12]:

	Hours	Scores
count	11.000000	11.000000
mean	5.245455	50.818182
std	2.773937	27.661591
min	1.500000	20.000000
25%	2.950000	26.000000
50%	5.100000	47.000000
75%	8.000000	78.000000
max	9.200000	88.000000

In [13]:
```
dataset.plot(x='Hours', y='Scores', style='o')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



In [14]:
```
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 1].values
```

In [15]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

In [16]:
```
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Out[16]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [17]:
```
print(regressor.intercept_)
```

```
2.9807407407407425
```

In [18]:
```
print(regressor.coef_)
```

```
[9.38207193]
```

In [19]:
```
y_pred = regressor.predict(X_test)
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df
```

Out[19]:

	Actual	Predicted
0	30	35.817992
1	25	28.312335
2	27	33.003371

In [20]:
```
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
Mean Absolute Error: 5.0445661120057315
Mean Squared Error: 26.953687195277723
Root Mean Squared Error: 5.191694058327949
```

Result:

Machine learning algorithm was implemented successfully.