Linear Regression

In [54]:
```
import pandas as pd
```

In [55]:
```
path_to_file = './student_scores.csv'
df = pd.read_csv(path_to_file)
```

In [56]:
```
df.head()
```

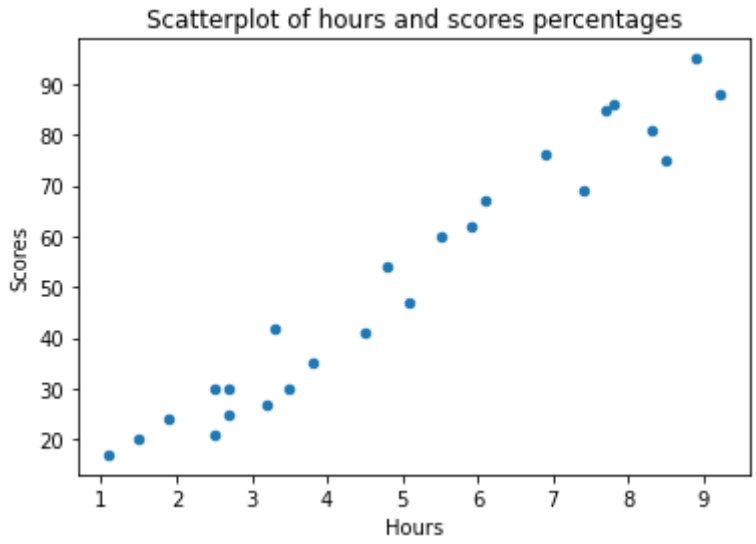
Out[56]:

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30

In [57]:
```
df.shape
```

Out[57]:
```
(25, 2)
```

In [58]:
```
df.plot.scatter(x='Hours', y='Scores', title='Scatterplot of hours and scores perce
```



In [59]:
```
print(df.corr())
```

```
            Hours    Scores
Hours    1.000000  0.976191
Scores   0.976191  1.000000
```

In [60]:
```
print(df.describe())
```

```
           Hours     Scores
count  25.000000  25.000000
mean    5.012000  51.480000
std     2.525094  25.286887
min     1.100000  17.000000
25%     2.700000  30.000000
50%     4.800000  47.000000
75%     7.400000  75.000000
max     9.200000  95.000000
```

In [61]:
```
Y = df['Scores'].values.reshape(-1, 1)
X = df['Hours'].values.reshape(-1, 1)
```

In [62]:
```
X
```

Out[62]:
```
array([[2.5],
       [5.1],
       [3.2],
       [8.5],
       [3.5],
       [1.5],
       [9.2],
       [5.5],
       [8.3],
       [2.7],
       [7.7],
       [5.9],
       [4.5],
       [3.3],
       [1.1],
       [8.9],
       [2.5],
       [1.9],
       [6.1],
       [7.4],
       [2.7],
       [4.8],
       [3.8],
       [6.9],
       [7.8]])
```

In [63]:
```
Y
```

Out[63]:
```
array([[21],
       [47],
       [27],
       [75],
       [30],
       [20],
       [88],
       [60],
       [81],
       [25],
       [85],
       [62],
       [41],
       [42],
       [17],
       [95],
       [30],
       [24],
       [67],
       [69],
       [30],
       [54],
       [35],
       [76],
       [86]], dtype=int64)
```

In [64]:
```
from sklearn.model_selection import train_test_split
```

In [65]:
```
SEED = 85
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_s
```

In [66]:
```
print(X_train)
print(Y_train)
```

```
[[2.5]
 [4.8]
 [6.9]
 [2.7]
 [9.2]
 [5.5]
 [5.9]
 [8.5]
 [1.1]
 [2.7]
 [8.3]
 [5.1]
 [8.9]
 [1.5]
 [3.8]
 [7.4]
 [3.5]
 [3.3]
 [2.5]
 [7.7]]
[[30]
 [54]
 [76]
 [25]
 [88]
 [60]
 [62]
 [75]
 [17]
 [30]
 [81]
 [47]
 [95]
 [20]
 [35]
 [69]
 [30]
 [42]
 [21]
 [85]]
```

In [67]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
```

In [68]:
```
regressor.fit(X_train, Y_train)
```

Out[68]:
```
LinearRegression()
```

In [69]:
```
print(regressor.intercept_)
```

```
[3.55813277]
```

In [70]:
```
print(regressor.coef_)
```

```
[[9.53671262]]
```

In [71]:
```
def calc(slope, intercept, hours):
    return slope*hours+intercept

score = calc(regressor.coef_, regressor.intercept_, 9.5)
print(score)
```

```
[[94.15690265]]
```

In [72]:
```
score = regressor.predict([[9.5]])
print(score)
```

```
[[94.15690265]]
```

In [73]:
```
Y_pred = regressor.predict(X_test)
```

In [74]:
```
df_preds = pd.DataFrame({'Actual': Y_test.squeeze(), 'Predicted':Y_pred.squeeze()})
print(df_preds)
```

```
   Actual  Predicted
0      27  34.075613
1      86  77.944491
2      41  46.473340
3      24  21.677887
4      67  61.732080
```

In [75]:
```
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np
```

In [76]:
```
mae = mean_absolute_error(Y_test, Y_pred)
mse = mean_squared_error(Y_test, Y_pred)
rmse = np.sqrt(mse)
```

In [77]:
```
print(f'Mean absolute error: {mae:.2f}')
print(f'Mean squared error: {mse:.2f}')
print(f'Root mean squared error: {rmse:.2f}')
```

```
Mean absolute error: 5.64
Mean squared error: 35.61
Root mean squared error: 5.97
```

In [78]:
```
#Train Error
regressor.score(X_train, Y_train)
```

Out[78]:
```
0.9555415361003867
```

In [79]:
```
#Test Error
regressor.score(X_test, Y_test)
```

Out[79]:
```
0.9378729367585895
```

Here we can see that training accuaracy is 94.91% and test accuracy is 96.78%.