In [61]:
```
#Linear Regression using scikitlearn and CuML, performance comparison and finding trado
```

In [62]:
```
import cudf
from cuml import make_regression, train_test_split
from cuml.linear_model import LinearRegression as cuLinearRegression
from cuml.metrics.regression import r2_score
from sklearn.linear_model import LinearRegression as skLinearRegression
```

In [63]:
```
#Creating datasset

samples = 2**19
features = 300
random_state = 42

X, y = make_regression(n_samples=samples, n_features=features, random_state=random_stat

X = cudf.DataFrame(X)
y = cudf.DataFrame(y)[0]

X_cu_train, X_cu_test, y_cu_train, y_cu_test = train_test_split(X, y, test_size=0.2, ra
```

In [64]:
```
#Convert the data into pandas
X_train = X_cu_train.to_pandas()
X_test = X_cu_test.to_pandas()
y_train = y_cu_train.to_pandas()
y_test = y_cu_test.to_pandas()
```

In [65]:
```
%%time
#Linear Regression Model using scikitlearn
ml1 = skLinearRegression(fit_intercept=True, normalize=True, n_jobs=-1)
ml1.fit(X_train, y_train)
```

```
CPU times: user 26.1 s, sys: 111 ms, total: 26.2 s
Wall time: 26.2 s
```

Out[65]:
```
LinearRegression(n_jobs=-1, normalize=True)
```

In [66]:
```
%%time
predict_ml1 = ml1.predict(X_test)
```

```
CPU times: user 28.9 ms, sys: 0 ns, total: 28.9 ms
Wall time: 27.7 ms
```

In [67]:
```
r2score_ml1 = r2_score(y_test, predict_ml1)
print(r2score_ml1)
```

```
1.0
```

In [68]:
```
%%time
#Linear Regression using CUML
```

```
cu_ml = cuLinearRegression(fit_intercept=True, normalize=True, algorithm='eig')
cu_ml.fit(X_cu_train, y_cu_train)
```

```
CPU times: user 170 ms, sys: 4.09 ms, total: 174 ms
Wall time: 174 ms
```

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

In [69]:
```
%%time
predict_cuml = cu_ml.predict(X_cu_test)
```

```
CPU times: user 21.4 ms, sys: 63 µs, total: 21.5 ms
Wall time: 20.6 ms
```

In [70]:
```
r2score_cuml = r2_score(y_cu_test, predict_cuml)
print(r2score_cuml)
```

```
1.0
```