Multiple Linear Regression

```
# from google.colab import files
# up = files.upload()
```

import dataset

```
import pandas as pd
df = pd.read_csv('dataset.csv')
df = df[['A', 'B', 'C', 'T']]
df.head()
```

	A	B	C	T
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244

```
# df.info()
```

cleaning

```
# clean the data
```

encoding

```
# encode the data
```

define x , y

```
import numpy as np
x = np.array(df[['A', 'B', 'C']])
y = np.array(df['T'])

# x = df[['ENGINESIZE', 'CYLINDERS', 'FUELCONSUMPTION_COMB']].values
# y = df['CO2EMISSIONS'].values

y[:5]
```

```
array([196, 221, 136, 255, 244], dtype=int64)
```

spliting

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25)
```

scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler().fit(x_train)
```

```
x_train = sc.transform(x_train)
x_test = sc.transform(x_test)
```

fit train data

```
# def param
# fit_intercept=True, copy_X=True, n_jobs=None, positive=False
```

```
from sklearn.linear_model import LinearRegression
mlr = LinearRegression()
mlr.fit(x_train, y_train)
```

```
print(mlr.intercept_)
print(mlr.coef_)
```

```
256.2187499999999
[16.20130859 12.55446815 33.38071933]
```

```
### K-fold cross validation

# from sklearn.linear_model import LinearRegression
# from sklearn.model_selection import GridSearchCV

# parameters = {
#     '': [],
#     '': []
# }

# lr = LinearRegression()
# gs = GridSearchCV(estimator=lr, param_grid=parameters, cv=5)

# gs.fit(x_train, y_train)

# best_params = gs.best_params_
# print(best_params)
```

predict test data

```
yhat_test = mlr.predict(x_test)
```

evaluate the model

```
from sklearn.metrics import r2_score
print("r2-score: %0.2f" % r2_score(y_test, yhat_test))
```

```
r2-score: 0.87
```

```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
print(f"MSE: {mean_squared_error(y_test, yhat_test)}")
print(f"MAE: {mean_absolute_error(y_test, yhat_test)}")
```

```
MSE: 534.3107477423459
MAE: 16.910810786461692
```

predict new data

```
mlr.predict(sc.transform([[2, 4, 8.5]]))
```

```
array([198.32166017])
```

save the model

```
# import joblib
# joblib.dump(mlr, 'mlr_model.pkl')
```

load the model

```
# import joblib
# mlr = joblib.load('mlr_model.pkl')
```