

LAB 8 – L1 Regularization (Lasso)

2403A52006

MD.Mustafa

✓ Import Libraries

```
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression, Lasso
from sklearn.metrics import mean_squared_error, r2_score
```

✓ Upload and Load Dataset

```
df = pd.read_csv("CO2 Emissions_Canada.csv")

df.head()
```

	Make	Model	Vehicle Class	Engine Size(L)	Cylinders	Transmission	Fuel Type	Consum City (
0	ACURA	ILX	COMPACT	2.0	4	AS5	Z	
1	ACURA	ILX	COMPACT	2.4	4	M6	Z	
2	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	Z	
3	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	Z	
4	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	Z	

Next steps:

[Generate code with df](#)[New interactive sheet](#)

✓ Select Required Features

```
X = df[['Engine Size(L)',
        'Cylinders',
```

```
'Fuel Consumption Comb (L/100 km)']]]  
  
y = df['CO2 Emissions(g/km)']
```

✓ Train Test Split

```
X_train, X_test, y_train, y_test = train_test_split(  
    X, y, test_size=0.2, random_state=42  
)
```

✓ Apply StandardScaler

```
scaler = StandardScaler()  
  
X_train_scaled = scaler.fit_transform(X_train)  
X_test_scaled = scaler.transform(X_test)
```

✓ Linear Regression

```
lin_model = LinearRegression()  
lin_model.fit(X_train_scaled, y_train)  
  
y_pred_lin = lin_model.predict(X_test_scaled)  
  
mse_lin = mean_squared_error(y_test, y_pred_lin)  
r2_lin = r2_score(y_test, y_pred_lin)  
  
print(" ♦ Linear Regression Results")  
print("MSE:", round(mse_lin, 2))  
print("R2 Score:", round(r2_lin, 4))
```

```
♦ Linear Regression Results  
MSE: 421.92  
R2 Score: 0.8773
```

✓ Lasso with GridSearchCV

```
lasso = Lasso(max_iter=10000)  
  
param_grid = {  
    'alpha': [0.0001, 0.001, 0.01, 0.1, 1, 10, 100]  
}  
  
grid = GridSearchCV(  
    lasso,  
    param_grid,  
    cv=5,
```

```
        scoring='r2'
    )

    grid.fit(X_train_scaled, y_train)

    best_lasso = grid.best_estimator_

    print("🏆 Best Alpha:", grid.best_params_)
```

🏆 Best Alpha: {'alpha': 0.01}

✓ Evaluate Best Lasso Model

```
y_pred_lasso = best_lasso.predict(X_test_scaled)

mse_lasso = mean_squared_error(y_test, y_pred_lasso)
r2_lasso = r2_score(y_test, y_pred_lasso)

print(" ♦ Lasso Regression Results")
print("MSE:", round(mse_lasso, 2))
print("R2 Score:", round(r2_lasso, 4))
```

♦ Lasso Regression Results
MSE: 421.93
R2 Score: 0.8773

✓ Compare Linear vs Lasso

```
print("\n📊 Model Comparison")

print("\nLinear Regression:")
print("MSE:", round(mse_lin, 2))
print("R2:", round(r2_lin, 4))

print("\nLasso Regression:")
print("MSE:", round(mse_lasso, 2))
print("R2:", round(r2_lasso, 4))
```

📊 Model Comparison



Linear Regression:
MSE: 421.92
R2: 0.8773

Lasso Regression:
MSE: 421.93
R2: 0.8773

✓ Show Coefficients

```
coef_df = pd.DataFrame({  
    "Feature": X.columns,  
    "Linear Coef": lin_model.coef_,  
    "Lasso Coef": best_lasso.coef_  
})
```

coef_df

	Feature	Linear Coef	Lasso Coef	
0	Engine Size(L)	7.605027	7.621766	
1	Cylinders	11.702725	11.686923	
2	Fuel Consumption Comb (L/100 km)	38.330321	38.319008	

Next steps:

[Generate code with coef_df](#)[New interactive sheet](#)