

```
import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.metrics import mean_squared_error
```

```
df = pd.read_csv('/content/drive/MyDrive/Concept and Technology of AI/student.csv')
df.head()
```

	Math	Reading	Writing
0	48	68	63
1	62	81	72
2	79	80	78
3	76	83	79
4	59	64	62

```
X = df.drop('Writing', axis=1)
y = df['Writing']
```

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

```
lin_reg = LinearRegression()
lin_reg.fit(X_train, y_train)
```

LinearRegression ⓘ ?

```
LinearRegression()
```

```
print("=== Baseline Linear Regression ===")
print("Coefficients:", lin_reg.coef_)
print("Intercept:", lin_reg.intercept_)
```

```
=== Baseline Linear Regression ===
Coefficients: [0.09212476 0.91363985]
Intercept: -1.538927419707349
```

```
y_train_pred = lin_reg.predict(X_train)
y_test_pred = lin_reg.predict(X_test)
```

```
print("Train MSE:", mean_squared_error(y_train, y_train_pred))
print("Test MSE:", mean_squared_error(y_test, y_test_pred))
```

```
Train MSE: 20.42671379648373
Test MSE: 22.92523648341902
```

```
alpha_grid = {"alpha": np.logspace(-3, 0, 13)} # 0.001 ... 1
```

```
ridge = Ridge(random_state=42)
lasso = Lasso(random_state=42, max_iter=10000)
```

```
ridge_cv = GridSearchCV(
    ridge, alpha_grid, cv=5, scoring="neg_mean_squared_error", n_jobs=-1
```

```
)
lasso_cv = GridSearchCV(
    lasso, alpha_grid, cv=5, scoring="neg_mean_squared_error", n_jobs=-1
)
```

```
ridge_cv.fit(X_train, y_train)
lasso_cv.fit(X_train, y_train)
```



```
print("\n=== Hyperparameter Tuning Results ===")
print("Best Ridge alpha:", ridge_cv.best_params_["alpha"])
print("Best Ridge CV MSE:", -ridge_cv.best_score_)
print("Best Lasso alpha:", lasso_cv.best_params_["alpha"])
print("Best Lasso CV MSE:", -lasso_cv.best_score_)
```

```
=== Hyperparameter Tuning Results ===
Best Ridge alpha: 1.0
Best Ridge CV MSE: 20.521736849527862
Best Lasso alpha: 0.005623413251903491
Best Lasso CV MSE: 20.521654529872755
```

```
best_ridge = ridge_cv.best_estimator_
best_lasso = lasso_cv.best_estimator_
```

```
ridge_train_pred = best_ridge.predict(X_train)
ridge_test_pred = best_ridge.predict(X_test)
lasso_train_pred = best_lasso.predict(X_train)
lasso_test_pred = best_lasso.predict(X_test)
```

```
print("\n=== Ridge (L2) with best alpha ===")
print("Coefficients:", best_ridge.coef_)
print("Train MSE:", mean_squared_error(y_train, ridge_train_pred))
print("Test MSE:", mean_squared_error(y_test, ridge_test_pred))
```

```
=== Ridge (L2) with best alpha ===
Coefficients: [0.09213514 0.91362602]
Train MSE: 20.426713811078994
Test MSE: 22.925202313725666
```

```
print("\n=== Lasso (L1) with best alpha ===")
print("Coefficients:", best_lasso.coef_)
print("Train MSE:", mean_squared_error(y_train, lasso_train_pred))
print("Test MSE:", mean_squared_error(y_test, lasso_test_pred))
```

```
=== Lasso (L1) with best alpha ===
Coefficients: [0.09220777 0.91354592]
Train MSE: 20.426714484075706
Test MSE: 22.924964900912222
```

Start coding or [generate](#) with AI.

