Hyperparameter Tuning Explanation (LassoCV, RidgeCV, ElasticNetCV)

What is Hyperparameter Tuning?

Hyperparameters are settings chosen before training the model. Tuning means trying multiple combinations to find the best one.

Analogy: Choosing bat size, grip tightness, stance before batting in cricket.

Regularization Models – Used to reduce overfitting.

Analogy: In exams, overfitting = writing irrelevant extra info. Regularization = write only relevant.

1) LassoCV (L1 Regularization)

- Removes unnecessary features (coefficients can become 0).
- CV finds best alpha.

Analogy: Packing a bag – throw out unused items.

Best when only some features matter.

Parameter: alpha.

2) RidgeCV (L2 Regularization)

- Shrinks coefficients but doesn’t remove them.
- CV finds best alpha.

Analogy: Roll clothes to pack tightly – keep all features but reduce influence.

Best when all features matter.

Parameter: alpha.

3) ElasticNetCV (Combination of L1 + L2)

- Partially removes and shrinks features.
- Parameters: alpha, l1_ratio (0 to 1).

Analogy: Throw unnecessary items and pack rest tightly.

Best when feature relationships are complex.

Comparison Table:

Model	Feature Removal	Feature Shrink	Best For	Hyperparameters
LassoCV	Yes	Some	Sparse features	alpha
RidgeCV	No	Yes	All features useful	alpha
ElasticNetCV	Yes (partial)	Yes	Balanced use case	alpha, l1_ratio

Final Note

These CV models automatically tune parameters using cross-validation.

Analogy: Practice matches help choose best batting stance.

Example Code:

```
from sklearn.linear_model import LassoCV, RidgeCV, ElasticNetCV
```

```
lasso = LassoCV(cv=5).fit(X_train, y_train)
ridge = RidgeCV(cv=5).fit(X_train, y_train)
elastic = ElasticNetCV(cv=5, l1_ratio=[0.2, 0.5, 0.8]).fit(X_train, y_train)

print(lasso.alpha_, ridge.alpha_, elastic.alpha_, elastic.l1_ratio_)
```