**Linear Regression Implementation**

Parameters-

1. **fit\_intercept : *bool, default=True***

Whether to calculate the intercept for this model. If set to False, no intercept will be used in calculations (i.e. data is expected to be centered).

2. normalize : bool, default=False  
    This parameter is ignored when fit\_intercept is set to False. If True, the regressors X will be normalized before regression by subtracting the mean and dividing by the l2-norm. If you wish to standardize, please use ~sklearn.preprocessing.StandardScaler before calling fit on an estimator with normalize=False.

Ideally we should use the fit\_intercept also plays important role.

Random\_state-Train Test Split

Controls the shuffling applied to the data before applying the split. Pass an int for reproducible output across multiple function calls.

IMP Interview questions-

A) Do we have to standardize or normalize the data in linear regression?

🡪 In linear regression gradient descent is formed so we are supposed to standardize the data. For standardization we use standard scalar.

Std value=(value-mean)/std deviation

If we don’t do standardization then independent feature can have any high value which will take long time to reach global minima.

B) How can we do hyperparameter tuning in linear regression?

🡪 As we could see very few parameters are there in linear regression either we can directly go for ridge, lasso, elasticNet which are upgrades of linear regression. Or we can use cross validation with pure models.

C) Why we do transform on test data but fit\_transform on training data?

🡪 fit\_transform is applied to test data to it will calculate mean and std dev for train data and transform it. Now same mean and std dev should be used for test data so only transform will be done. If we do fit on test data then information about test data will be leaked and this should not happen.