Random State

In Sklearn's Train Test Split or any other Algorithm, you will find an argument called 'random_state'. Its function is to make the results reproducible.

Sklearn's train_test_split splits arrays or matrices into **random** train and test subsets. When you run the algorithm without specifying a random_state, you will get a different result every time, this is an expected behaviour. Random State controls the shuffling applied to the data before applying the split

In unsupervised algorithms like KMeans, or Tree Based/Ensemble methods like DT/RF, using random_state is preferred to get reproducible results. All these algorithms will give different results on every run without random_state.

Train Test Split Example:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
```

KMeans Random State Example:

```
kmeans = KMeans(n_clusters=2, random_state=0)
```

Decision Tree and Random Forest Example:

```
clf = tree.DecisionTreeClassifier(random_state=0)

clf = RandomForestClassifier(random_state=0)
```

Similarly, you can check every algorithm's documentation here to see where to enter random_state.