Randomised search CV

#Randomized Search CV

1. Define all feature like below

```
# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 100, stop = 1200, num = 12)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(5, 30, num = 6)]
# max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10, 15, 100]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 5, 10]
```

2. Create a RandomGrid

```
# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf}
print(random_grid)
```

3. Then create a model where you tune

```
# First create the base model to tune
rf = RandomForestRegressor()
```

4. Apply Randomised Search Cv

```
# search across 100 different combinations
rf_random = RandomizedSearchCV( estimator = rf, param_distributions =
random_grid,scoring='neg_mean_squared_error', n_iter = 10, cv = 5, verbose=2, random_state=42,
n_jobs = 1)
```

5. Fit X_train And Y_train

```
rf_random.fit(X_train,y_train)
```