

```
In [17]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [18]: df = pd.read_csv('Final.csv')
df.head()
```

```
Out[18]:
```

	Item_Identifier	Item_Weight	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Item_Outlet_Sales	Outl
0	FDA15	9.30	0.922960	Dairy	249.8092	OUT049	1999	3735.1380	
1	DRC01	5.92	1.003057	Soft Drinks	48.2692	OUT018	2009	443.4228	
2	FDN15	17.50	0.831990	Meat	141.6180	OUT049	1999	2097.2700	
3	FDX07	19.20	0.750000	Fruits and Vegetables	182.0950	OUT010	1998	732.3800	
4	NCD19	8.93	0.666667	Household	53.8614	OUT013	1987	994.7052	

5 rows × 30 columns



```
In [19]: remove_cols = [
    'Item_Identifier',
    'Item_Type',
    'Outlet_Identifier',
    'Outlet_Establishment_Year'
]
df = df.drop(remove_cols,axis =1)
df.head()
```

Out[19]:

	Item_Weight	Item_Visibility	Item_MRP	Item_Outlet_Sales	Outlet_Years	Item_Fat_Content_1	Item_Fat_Content_2	Outlet_Location_Type_1	Outlet_Location_Type_2
0	9.30	0.922960	249.8092	3735.1380	14	False	False	False	False
1	5.92	1.003057	48.2692	443.4228	4	False	True	False	False
2	17.50	0.831990	141.6180	2097.2700	14	False	False	False	False
3	19.20	0.750000	182.0950	732.3800	15	False	True	False	False
4	8.93	0.666667	53.8614	994.7052	26	True	False	False	False

5 rows × 26 columns



In [20]: `df.shape`

Out[20]: (8519, 26)

In [21]: `y = df.Item_Outlet_Sales.values`  
`X = df.drop('Item_Outlet_Sales',axis = 1)`

In [22]: `print(X.shape,y.shape)`

(8519, 25) (8519,)

In [23]: `from sklearn.ensemble import RandomForestRegressor`  
`rf = RandomForestRegressor()`

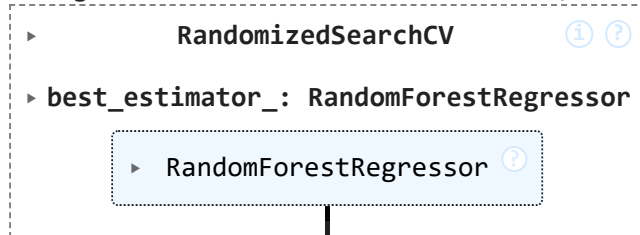
In [24]: `param = {`  
 `'max_depth':[3,6,9,12],`  
 `'n_estimators' : [10,50,100,200]`  
`}`

In [25]: `from sklearn.metrics import mean_squared_error,make_scorer`  
`from sklearn.model_selection import RandomizedSearchCV`  
`random_search=RandomizedSearchCV(rf,param_distributions=param,n_iter=5,scoring=make_scorer(mean_squared_error),n_jobs=-1,cv=5,`

In [26]: `random_search.fit(X,y)`

Fitting 5 folds for each of 5 candidates, totalling 25 fits

Out[26]:



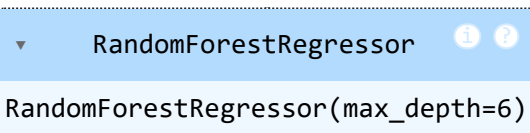
```
In [27]: means = random_search.cv_results_['mean_test_score']
params = random_search.cv_results_['params']
for mean, param in zip(means, params):
    print("%f with: %r" % (mean, param))
    if mean == min(means):
        print('Best parameters with the minimum Mean Square Error are:', param)
```

```
1199402.774718 with: {'n_estimators': 50, 'max_depth': 6}
1205259.865622 with: {'n_estimators': 200, 'max_depth': 9}
1274767.773600 with: {'n_estimators': 10, 'max_depth': 12}
1392180.998621 with: {'n_estimators': 100, 'max_depth': 3}
1198863.598698 with: {'n_estimators': 100, 'max_depth': 6}
Best parameters with the minimum Mean Square Error are: {'n_estimators': 100, 'max_depth': 6}
```

```
In [28]: rf = RandomForestRegressor(
    n_estimators=100, max_depth=6,
)
```

```
In [29]: rf.fit(X,y)
```

Out[29]:



```
In [30]: y_pred = rf.predict(X)
```

```
In [31]: from sklearn.metrics import r2_score, mean_squared_error

score = r2_score(y, y_pred)
```

```
print("Score:", score)
print("RMSE : %.4g" % np.sqrt(mean_squared_error(y, y_pred)))
```

Score: 0.6157037236877372

RMSE : 1058

```
In [32]: from sklearn.metrics import mean_squared_error, make_scorer, mean_absolute_error
from sklearn.model_selection import cross_val_score
cv_score = cross_val_score(rf, X, y, cv=20, scoring = make_scorer(mean_squared_error))
cv_score = np.sqrt(np.abs(cv_score))
print("\nModel Report")
print("MAE : %.4g" % np.sqrt(mean_absolute_error(y, y_pred)))
print("RMSE : %.4g" % np.sqrt(mean_squared_error(y, y_pred)))
print("CV Score : Mean - %.4g | Std - %.4g | Min - %.4g | Max - %.4g" % (np.mean(cv_score), np.std(cv_score), np.min(cv_score), np.max(cv_score)))
```

Model Report

MAE : 27.29

RMSE : 1058

CV Score : Mean - 1091 | Std - 46.87 | Min - 1020 | Max - 1209