```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from sklearn.preprocessing import StandardScaler  , LabelEncoder
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split , cross_val_score , GridSearchCV
from catboost  import CatBoostClassifier
```

```
data = pd.read_csv('data.csv')
```

```
data.head()
```

	Date	City	CO	NO2	SO2	O3	PM2.5	PM10	Type
0	2024-01-01 00:00:00+00:00	Moscow	208.0	15.9	13.2	44.0	8.6	9.4	Industrial
1	2024-01-01 01:00:00+00:00	Moscow	207.0	17.4	13.7	44.0	8.6	10.5	Industrial
2	2024-01-01 02:00:00+00:00	Moscow	217.0	19.0	15.5	43.0	10.4	12.9	Industrial
3	2024-01-01 03:00:00+00:00	Moscow	231.0	21.0	20.7	36.0	12.3	15.3	Industrial
4	2024-01-01 04:00:00+00:00	Moscow	263.0	34.5	27.2	27.0	13.6	20.0	Industrial

Next steps: View recommended plots | New interactive sheet

```
X = data.drop(columns=['Date' , 'City' , 'Type'])
Y = data['Type']
```

```
y_le = LabelEncoder()
Y = y_le.fit_transform(Y)
```

```
X_train , X_test , y_train , y_test = train_test_split(X , Y , test_size=0.3 , random_state=42)
```

```
X.describe()
```

	CO	NO2	SO2	O3	PM2.5	PM10
count	52704.000000	52704.000000	52704.000000	52704.000000	52704.000000	52704.000000
mean	508.030472	29.616492	22.387250	53.423668	32.931045	50.644980
std	692.274824	23.984409	34.285823	42.225556	43.833691	67.311502
min	0.000000	0.900000	0.000000	0.000000	0.300000	0.400000
25%	187.000000	11.000000	0.700000	26.000000	6.400000	9.400000
50%	268.000000	23.300000	10.500000	48.000000	14.800000	19.800000
75%	519.000000	42.200000	30.200000	69.000000	42.600000	68.350000
max	12876.000000	218.000000	497.800000	342.000000	459.100000	661.200000

```
model = CatBoostClassifier(
    silent=True,
    random_seed=42
)
```

```
param_grid = {
    'iterations': [ 500 ],
    'learning_rate': [0.01, 0.1],
    'depth': [4, 6, 8 ],

}
```

```
grid = GridSearchCV(estimator=model, param_grid=param_grid,
                    scoring='accuracy', cv=3, n_jobs=-1, verbose=1)
```

```
grid.fit(X_train, y_train)


print("Best Parameters:")
print(grid.best_params_)
```

```
Fitting 3 folds for each of 6 candidates, totalling 18 fits
Best Parameters:
{'depth': 8, 'iterations': 500, 'learning_rate': 0.1}
```

```
print("Best Score:")
print(grid.best_score_)
```

```
Best Score:
0.9883443292981638
```

```
best_param = grid.best_params_
```

```
cbg = CatBoostClassifier(**best_param)
```

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

```
495:    learn: 0.0073554        total: 10.9s    remaining: 88ms
496:    learn: 0.0073554        total: 10.9s    remaining: 66ms
497:    learn: 0.0073554        total: 10.9s    remaining: 43.9ms
498:    learn: 0.0073554        total: 11s      remaining: 22ms
499:    learn: 0.0073554        total: 11s      remaining: 0us
<catboost.core.CatBoostClassifier at 0x7d824a694650>
```

```
y_pred = cbg.predict(X_test)
```

```
print(r2_score(y_test , y_pred))
```

```
0.9612949457394157
```