```
import pandas as pd
import numpy as np
```

```
train =pd.read_csv('./data/4th-t2/train.csv')
test=pd.read_csv('data/4th-t2/test.csv')
```

```
train.shape, test.shape
```

```
((6665, 11), (2154, 10))
```

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6665 entries, 0 to 6664
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Gender           6665 non-null   object 
 1   Ever_Married     6665 non-null   object 
 2   Age              6665 non-null   int64  
 3   Graduated        6665 non-null   object 
 4   Profession       6665 non-null   object 
 5   Work_Experience  6665 non-null   float64
 6   Spending_Score   6665 non-null   object 
 7   Family_Size      6665 non-null   float64
 8   Var_1            6665 non-null   object 
dtypes: float64(2), int64(1), object(6)
memory usage: 468.8+ KB
```

```
train.head(2)
```

	ID	Gender	Ever_Married	Age	Graduated	Profession	Work_Experience	Spending_Score	Family_Size	Var_1
0	462809	Male	No	22	No	Healthcare	1.0	Low	4.0	Cat_4
1	466315	Female	Yes	67	Yes	Engineer	1.0	Low	1.0	Cat_6

```
test.head(2)
```

	ID	Gender	Ever_Married	Age	Graduated	Profession	Work_Experience	Spending_Score	Family_Size	Var_1
0	458989	Female	Yes	36	Yes	Engineer	0.0	Low	1.0	Cat_6
1	458994	Male	Yes	37	Yes	Healthcare	8.0	Average	4.0	Cat_6

```
train.pop('ID')
test_ID=test.pop('ID')
```

```
target=train.pop('Segmentation')
```

```
train.shape, test.shape
```

```
((6665, 9), (2154, 9))
```

```
num_cols= train.select_dtypes(['float64','int64']).columns
```

```
cat_cols=train.select_dtypes(['object']).columns
```

```
train= pd.get_dummies(train)
test= pd.get_dummies(test)
```

```
train.shape, test.shape
```

```
((6665, 28), (2154, 28))
```

```
train.columns.equals(test.columns)
```

```
True
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
rf=RandomForestClassifier(random_state=42, n_estimators=500, max_depth=6)
```

```
rf.fit(train,target)
```

```
▾ RandomForestClassifier
RandomForestClassifier(max_depth=6, n_estimators=500, random_state=42)
```

```
pred=rf.predict(test)
```

```
# 교차 검증  이 문제의 경우 test 의 target 값이 주어지지 않아서 train 의 교차검증을 할 수 밖에 없다
from sklearn.model_selection import cross_val_score
scores = cross_val_score(rf, train, target, scoring='f1_macro', cv=5)
print(scores)
print(scores.mean())
```

```
[0.5203876  0.50497926 0.50448728 0.52940535 0.50058356]
0.5119686102517507
```