In [35]:
```
import numpy as np
import pandas as pd
import random
random.seed(42)
np.random.seed(42)
```

In [36]:
```
df_train = pd.read_csv('train.csv')
df_train.shape
```

Out[36]:
```
(891, 12)
```

In [37]:
```
df_train.head()
```

Out[37]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500

In [38]:
```
X = df_train.drop(['Survived', 'PassengerId', 'Name', 'Ticket', 'Cabin'], axis=1)
y = df_train['Survived']
```

In [39]:
```
num_cols = X.select_dtypes(include=['int64', 'float64']).columns.tolist()
cat_cols = X.select_dtypes(include=['object']).columns.tolist()
```

In [40]:
```
from sklearn.model_selection import train_test_split

X_train_val, X_test, y_train_val, y_test = train_test_split(X, y, test_size=0.2, ra
X_train, X_val, y_train, y_val = train_test_split(X_train_val, y_train_val, test_si
```

In [41]:
```
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
```

In [42]:
```
num_pipeline = Pipeline(steps=[
    ("imputer", SimpleImputer(strategy="median")),
    ("scaler", StandardScaler())
])

cat_pipeline = Pipeline(steps=[
    ("imputer", SimpleImputer(strategy="most_frequent")),
    ("encoder", OneHotEncoder(handle_unknown="ignore"))
])

preprocessor = ColumnTransformer(transformers=[
    ("num", num_pipeline, num_cols),
    ("cat", cat_pipeline, cat_cols)
], remainder="drop")
```

In [43]:
```
params = {
    "n_estimators" : [100,200,300],
    "max_depth" : [None, 3,5,8],
    "max_leaf_nodes" : [2,4,8],
    "max_features" : [None, "sqrt", "log2"],
    "n_jobs" : [-1]
}

from sklearn.model_selection import ParameterGrid
from sklearn.metrics import f1_score
scores = []


for config in ParameterGrid(params):
    model = Pipeline(steps=[
        ("preprocessor", preprocessor),
        ("RF", RandomForestClassifier(**config, random_state=42))
    ])

    model.fit(X_train,y_train)
    y_pred = model.predict(X_val)
    score = f1_score(y_val, y_pred)
    print(config, "f1 --> ", score)
    scores.append((score))
```

```
{'max_depth': None, 'max_features': None, 'max_leaf_nodes': 2, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': None, 'max_features': None, 'max_leaf_nodes': 2, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': None, 'max_features': None, 'max_leaf_nodes': 2, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': None, 'max_features': None, 'max_leaf_nodes': 4, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.7580645161290323
{'max_depth': None, 'max_features': None, 'max_leaf_nodes': 4, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.7642276422764228
{'max_depth': None, 'max_features': None, 'max_leaf_nodes': 4, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.7642276422764228
{'max_depth': None, 'max_features': None, 'max_leaf_nodes': 8, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.7716535433070866
{'max_depth': None, 'max_features': None, 'max_leaf_nodes': 8, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.784
{'max_depth': None, 'max_features': None, 'max_leaf_nodes': 8, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.784
{'max_depth': None, 'max_features': 'sqrt', 'max_leaf_nodes': 2, 'n_estimators': 10
0, 'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': None, 'max_features': 'sqrt', 'max_leaf_nodes': 2, 'n_estimators': 20
0, 'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': None, 'max_features': 'sqrt', 'max_leaf_nodes': 2, 'n_estimators': 30
0, 'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': None, 'max_features': 'sqrt', 'max_leaf_nodes': 4, 'n_estimators': 10
0, 'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': None, 'max_features': 'sqrt', 'max_leaf_nodes': 4, 'n_estimators': 20
0, 'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': None, 'max_features': 'sqrt', 'max_leaf_nodes': 4, 'n_estimators': 30
0, 'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': None, 'max_features': 'sqrt', 'max_leaf_nodes': 8, 'n_estimators': 10
0, 'n_jobs': -1} f1 -->  0.768
{'max_depth': None, 'max_features': 'sqrt', 'max_leaf_nodes': 8, 'n_estimators': 20
0, 'n_jobs': -1} f1 -->  0.7741935483870968
{'max_depth': None, 'max_features': 'sqrt', 'max_leaf_nodes': 8, 'n_estimators': 30
0, 'n_jobs': -1} f1 -->  0.7741935483870968
{'max_depth': None, 'max_features': 'log2', 'max_leaf_nodes': 2, 'n_estimators': 10
0, 'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': None, 'max_features': 'log2', 'max_leaf_nodes': 2, 'n_estimators': 20
0, 'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': None, 'max_features': 'log2', 'max_leaf_nodes': 2, 'n_estimators': 30
0, 'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': None, 'max_features': 'log2', 'max_leaf_nodes': 4, 'n_estimators': 10
0, 'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': None, 'max_features': 'log2', 'max_leaf_nodes': 4, 'n_estimators': 20
0, 'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': None, 'max_features': 'log2', 'max_leaf_nodes': 4, 'n_estimators': 30
0, 'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': None, 'max_features': 'log2', 'max_leaf_nodes': 8, 'n_estimators': 10
0, 'n_jobs': -1} f1 -->  0.768
{'max_depth': None, 'max_features': 'log2', 'max_leaf_nodes': 8, 'n_estimators': 20
0, 'n_jobs': -1} f1 -->  0.7741935483870968
{'max_depth': None, 'max_features': 'log2', 'max_leaf_nodes': 8, 'n_estimators': 30
0, 'n_jobs': -1} f1 -->  0.7741935483870968
{'max_depth': 3, 'max_features': None, 'max_leaf_nodes': 2, 'n_estimators': 100, 'n_
jobs': -1} f1 -->  0.7401574803149606
```

```
{'max_depth': 3, 'max_features': None, 'max_leaf_nodes': 2, 'n_estimators': 200, 'n_
jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 3, 'max_features': None, 'max_leaf_nodes': 2, 'n_estimators': 300, 'n_
jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 3, 'max_features': None, 'max_leaf_nodes': 4, 'n_estimators': 100, 'n_
jobs': -1} f1 -->  0.7580645161290323
{'max_depth': 3, 'max_features': None, 'max_leaf_nodes': 4, 'n_estimators': 200, 'n_
jobs': -1} f1 -->  0.7642276422764228
{'max_depth': 3, 'max_features': None, 'max_leaf_nodes': 4, 'n_estimators': 300, 'n_
jobs': -1} f1 -->  0.7642276422764228
{'max_depth': 3, 'max_features': None, 'max_leaf_nodes': 8, 'n_estimators': 100, 'n_
jobs': -1} f1 -->  0.7936507936507936
{'max_depth': 3, 'max_features': None, 'max_leaf_nodes': 8, 'n_estimators': 200, 'n_
jobs': -1} f1 -->  0.7936507936507936
{'max_depth': 3, 'max_features': None, 'max_leaf_nodes': 8, 'n_estimators': 300, 'n_
jobs': -1} f1 -->  0.7936507936507936
{'max_depth': 3, 'max_features': 'sqrt', 'max_leaf_nodes': 2, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 3, 'max_features': 'sqrt', 'max_leaf_nodes': 2, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 3, 'max_features': 'sqrt', 'max_leaf_nodes': 2, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 3, 'max_features': 'sqrt', 'max_leaf_nodes': 4, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 3, 'max_features': 'sqrt', 'max_leaf_nodes': 4, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 3, 'max_features': 'sqrt', 'max_leaf_nodes': 4, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 3, 'max_features': 'sqrt', 'max_leaf_nodes': 8, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 3, 'max_features': 'sqrt', 'max_leaf_nodes': 8, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.7619047619047619
{'max_depth': 3, 'max_features': 'sqrt', 'max_leaf_nodes': 8, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.7619047619047619
{'max_depth': 3, 'max_features': 'log2', 'max_leaf_nodes': 2, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 3, 'max_features': 'log2', 'max_leaf_nodes': 2, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 3, 'max_features': 'log2', 'max_leaf_nodes': 2, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 3, 'max_features': 'log2', 'max_leaf_nodes': 4, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 3, 'max_features': 'log2', 'max_leaf_nodes': 4, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 3, 'max_features': 'log2', 'max_leaf_nodes': 4, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 3, 'max_features': 'log2', 'max_leaf_nodes': 8, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 3, 'max_features': 'log2', 'max_leaf_nodes': 8, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.7619047619047619
{'max_depth': 3, 'max_features': 'log2', 'max_leaf_nodes': 8, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.7619047619047619
{'max_depth': 5, 'max_features': None, 'max_leaf_nodes': 2, 'n_estimators': 100, 'n_
jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 5, 'max_features': None, 'max_leaf_nodes': 2, 'n_estimators': 200, 'n_
jobs': -1} f1 -->  0.7401574803149606
```

```
{'max_depth': 5, 'max_features': None, 'max_leaf_nodes': 2, 'n_estimators': 300, 'n_
jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 5, 'max_features': None, 'max_leaf_nodes': 4, 'n_estimators': 100, 'n_
jobs': -1} f1 -->  0.7580645161290323
{'max_depth': 5, 'max_features': None, 'max_leaf_nodes': 4, 'n_estimators': 200, 'n_
jobs': -1} f1 -->  0.7642276422764228
{'max_depth': 5, 'max_features': None, 'max_leaf_nodes': 4, 'n_estimators': 300, 'n_
jobs': -1} f1 -->  0.7642276422764228
{'max_depth': 5, 'max_features': None, 'max_leaf_nodes': 8, 'n_estimators': 100, 'n_
jobs': -1} f1 -->  0.7716535433070866
{'max_depth': 5, 'max_features': None, 'max_leaf_nodes': 8, 'n_estimators': 200, 'n_
jobs': -1} f1 -->  0.784
{'max_depth': 5, 'max_features': None, 'max_leaf_nodes': 8, 'n_estimators': 300, 'n_
jobs': -1} f1 -->  0.784
{'max_depth': 5, 'max_features': 'sqrt', 'max_leaf_nodes': 2, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 5, 'max_features': 'sqrt', 'max_leaf_nodes': 2, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 5, 'max_features': 'sqrt', 'max_leaf_nodes': 2, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 5, 'max_features': 'sqrt', 'max_leaf_nodes': 4, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 5, 'max_features': 'sqrt', 'max_leaf_nodes': 4, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 5, 'max_features': 'sqrt', 'max_leaf_nodes': 4, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 5, 'max_features': 'sqrt', 'max_leaf_nodes': 8, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.768
{'max_depth': 5, 'max_features': 'sqrt', 'max_leaf_nodes': 8, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.784
{'max_depth': 5, 'max_features': 'sqrt', 'max_leaf_nodes': 8, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.7741935483870968
{'max_depth': 5, 'max_features': 'log2', 'max_leaf_nodes': 2, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 5, 'max_features': 'log2', 'max_leaf_nodes': 2, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 5, 'max_features': 'log2', 'max_leaf_nodes': 2, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 5, 'max_features': 'log2', 'max_leaf_nodes': 4, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 5, 'max_features': 'log2', 'max_leaf_nodes': 4, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 5, 'max_features': 'log2', 'max_leaf_nodes': 4, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 5, 'max_features': 'log2', 'max_leaf_nodes': 8, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.768
{'max_depth': 5, 'max_features': 'log2', 'max_leaf_nodes': 8, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.784
{'max_depth': 5, 'max_features': 'log2', 'max_leaf_nodes': 8, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.7741935483870968
{'max_depth': 8, 'max_features': None, 'max_leaf_nodes': 2, 'n_estimators': 100, 'n_
jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 8, 'max_features': None, 'max_leaf_nodes': 2, 'n_estimators': 200, 'n_
jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 8, 'max_features': None, 'max_leaf_nodes': 2, 'n_estimators': 300, 'n_
jobs': -1} f1 -->  0.7401574803149606
```

```
{'max_depth': 8, 'max_features': None, 'max_leaf_nodes': 4, 'n_estimators': 100, 'n_
jobs': -1} f1 -->  0.7580645161290323
{'max_depth': 8, 'max_features': None, 'max_leaf_nodes': 4, 'n_estimators': 200, 'n_
jobs': -1} f1 -->  0.7642276422764228
{'max_depth': 8, 'max_features': None, 'max_leaf_nodes': 4, 'n_estimators': 300, 'n_
jobs': -1} f1 -->  0.7642276422764228
{'max_depth': 8, 'max_features': None, 'max_leaf_nodes': 8, 'n_estimators': 100, 'n_
jobs': -1} f1 -->  0.7716535433070866
{'max_depth': 8, 'max_features': None, 'max_leaf_nodes': 8, 'n_estimators': 200, 'n_
jobs': -1} f1 -->  0.784
{'max_depth': 8, 'max_features': None, 'max_leaf_nodes': 8, 'n_estimators': 300, 'n_
jobs': -1} f1 -->  0.784
{'max_depth': 8, 'max_features': 'sqrt', 'max_leaf_nodes': 2, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 8, 'max_features': 'sqrt', 'max_leaf_nodes': 2, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 8, 'max_features': 'sqrt', 'max_leaf_nodes': 2, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 8, 'max_features': 'sqrt', 'max_leaf_nodes': 4, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 8, 'max_features': 'sqrt', 'max_leaf_nodes': 4, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 8, 'max_features': 'sqrt', 'max_leaf_nodes': 4, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 8, 'max_features': 'sqrt', 'max_leaf_nodes': 8, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.768
{'max_depth': 8, 'max_features': 'sqrt', 'max_leaf_nodes': 8, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.7741935483870968
{'max_depth': 8, 'max_features': 'sqrt', 'max_leaf_nodes': 8, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.7741935483870968
{'max_depth': 8, 'max_features': 'log2', 'max_leaf_nodes': 2, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 8, 'max_features': 'log2', 'max_leaf_nodes': 2, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 8, 'max_features': 'log2', 'max_leaf_nodes': 2, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.7401574803149606
{'max_depth': 8, 'max_features': 'log2', 'max_leaf_nodes': 4, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 8, 'max_features': 'log2', 'max_leaf_nodes': 4, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 8, 'max_features': 'log2', 'max_leaf_nodes': 4, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.746031746031746
{'max_depth': 8, 'max_features': 'log2', 'max_leaf_nodes': 8, 'n_estimators': 100,
'n_jobs': -1} f1 -->  0.768
{'max_depth': 8, 'max_features': 'log2', 'max_leaf_nodes': 8, 'n_estimators': 200,
'n_jobs': -1} f1 -->  0.7741935483870968
{'max_depth': 8, 'max_features': 'log2', 'max_leaf_nodes': 8, 'n_estimators': 300,
'n_jobs': -1} f1 -->  0.7741935483870968
```

In [44]:
```
best_score = max(scores)
print(best_score)
best_index = scores.index(best_score)
best_config = list(ParameterGrid(params))[best_index]
print(best_config)
```

```
0.7936507936507936
{'max_depth': 3, 'max_features': None, 'max_leaf_nodes': 8, 'n_estimators': 100, 'n_
jobs': -1}
```

In [45]:
```
best_model = Pipeline(steps=[
    ("process", preprocessor),
    ("model", RandomForestClassifier(**best_config, random_state=42))
])

best_model.fit(X_train_val, y_train_val)
y_test_pred = best_model.predict(X_test)
score = f1_score(y_test,y_test_pred)
print(score)
```

```
0.7391304347826086
```