

```
In [ ]: from sklearn.neighbors import KNeighborsClassifier
        from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score,
        from sklearn.metrics import classification_report, confusion_matrix
        from sklearn.datasets import make_hastie_10_2
        from sklearn.ensemble import GradientBoostingClassifier
        from sklearn.svm import SVC

        import pandas as pd
        import warnings
        warnings.filterwarnings('ignore')
```

```
In [ ]: df = pd.read_csv("employee_churn_data_cleaned.csv")
        df
```

```
Out [ ]:
```

	department_0	department_1	department_2	department_3	department_4	departme
0	0.0	0.0	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	0.0	
2	0.0	1.0	0.0	0.0	0.0	
3	0.0	0.0	1.0	0.0	0.0	
4	0.0	0.0	1.0	0.0	0.0	
...	
9535	0.0	0.0	0.0	0.0	0.0	
9536	0.0	0.0	1.0	0.0	0.0	
9537	0.0	0.0	0.0	0.0	0.0	
9538	0.0	0.0	0.0	0.0	0.0	
9539	0.0	0.0	1.0	0.0	0.0	

9540 rows × 19 columns



```
In [ ]: X = df.loc[:, df.columns != 'left']
        y = df['left']
```

```
In [ ]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

```
In [ ]: grid = {
        # "n_neighbors": [5, 10, 20],
        "n_neighbors": [20],
        "algorithm": ['auto'],
        # "leaf_size": [10, 20, 30, 40, 50],
        "leaf_size": [10],
        # "p": [0.01, 0.1, 1, 2, 10, 100],
        "p": [0.01, 0.1],
    }
```

```
gcv = GridSearchCV(KNeighborsClassifier(), grid)
gcv.fit(X_train,y_train)
print(cross_val_score(gcv,X_train,y_train,cv=5))
print(gcv.score(X_test,y_test))
```

```
[0.83458084 0.82185629 0.83383234 0.82696629 0.8329588 ]
0.827393431167016
```

```
In [ ]: grid = {
        "gamma" : ["scale","auto"]
      }

      svc = GridSearchCV(SVC(),grid)
      svc.fit(X_train,y_train)
      print(cross_val_score(svc,X_train,y_train,cv=5))
      print(svc.score(X_test,y_test))
```

```
[0.84056886 0.83383234 0.84206587 0.83220974 0.8411985 ]
0.8287910552061496
```

```
In [ ]: grid = {
        "n_estimators":[300,500,800],
        "learning_rate":[0.1,0.5],
        "max_depth" : [3,5]
      }

      gcv = GridSearchCV(GradientBoostingClassifier(),grid)
      gcv.fit(X_train, y_train)
      print(cross_val_score(gcv,X_train,y_train,cv=5))
      print(gcv.best_params_)
      print(gcv.best_score_)
```

```
[0.84505988 0.83982036 0.85479042 0.83520599 0.84269663]
{'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 500}
0.8441137948821458
```

```
In [ ]:
```