Model Tuning

Cross Validation
And
Grid Search

```
8 # Importing the dataset
 9 dataset = pd.read csv('Social Network Ads.csv')
10 X = dataset.iloc[:, [2, 3]].values
11 y = dataset.iloc[:, 4].values
12
13 # Splitting the dataset into the Training set and Test set
14 from sklearn.model selection import train test split
15 X train, X test, y train, y test = train_test_split(X, y, test_size = 0.25, random_state = 0)
16
17 # Feature Scaling
18 from sklearn.preprocessing import StandardScaler
19 sc = StandardScaler()
20 X train = sc.fit transform(X train)
21 X test = sc.transform(X test)
22
23 # Fitting Kernel SVM to the Training set
24 from sklearn.svm import SVC
25 classifier = SVC(kernel = 'rbf', random state = 0)
26 classifier.fit(X train, y train)
28 # Predicting the Test set results
29 y pred = classifier.predict(X test)
31 # Making the Confusion Matrix
32 from sklearn.metrics import confusion matrix
33 cm = confusion matrix(y test, y pred)
34
35 # Applying k-Fold Cross Validation
36 from sklearn.model selection import cross val score
37 accuracies = cross val score(estimator = classifier, X = X train, y = y train, cv = 10)
38 accuracies.mean()
39 accuracies.std()
```

Grid Search

```
8 # Importing the dataset
9 dataset = pd.read csv('Social Network Ads.csv')
10 X = dataset.iloc[:, [2, 3]].values
11 y = dataset.iloc[:, 4].values
12
13 # Splitting the dataset into the Training set and Test set
14 from sklearn.model selection import train test split
15 X train, X test, y train, y test = train test split(X, y, test size = 0.25, random state = 0)
17 # Feature Scaling
18 from sklearn.preprocessing import StandardScaler
19 sc = StandardScaler()
20 X train = sc.fit transform(X train)
21 X test = sc.transform(X test)
23 # Fitting Kernel SVM to the Training set
24 from sklearn.svm import SVC
25 classifier = SVC(kernel = 'rbf', random state = 0)
26 classifier.fit(X_train, y_train)
28 # Predicting the Test set results
29 y pred = classifier.predict(X test)
31 # Making the Confusion Matrix
32 from sklearn.metrics import confusion matrix
33 cm = confusion matrix(y test, y pred)
35 # Applying k-Fold Cross Validation
36 from sklearn.model selection import cross val score
37 accuracies = cross val score(estimator = classifier, X = X train, y = y train, cv = 10)
38 accuracies.mean()
39 accuracies.std()
40
41# Applying Grid Search to find the best model and the best parameters
42 from sklearn.model_selection import GridSearchCV
43 parameters = [{'C': [1, 10, 100, 1000], 'kernel': ['linear']},
                {'C': [1, 10, 100, 1000], 'kernel': ['rbf'], 'gamma': [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0
45 grid_search = GridSearchCV(estimator = classifier,
46
                              param grid = parameters,
47
                              scoring = 'accuracy',
48
                              cv = 10,
                              n_{jobs} = -1)
50 grid search = grid search.fit(X train, y train)
51 best accuracy = grid search.best score
52 best_parameters = grid_search.best_params
53
```

Grid Search

```
6 import pandas as pd
 8 # Importing the dataset
9 dataset = pd.read_csv('Social_Network_Ads.csv')
10 X = dataset.iloc[:, [2, 3]].values
11 y = dataset.iloc[:, 4].values
13 # Splitting the dataset into the Training set and Test set
14 from sklearn.model selection import train test split
15 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
16
17 # Feature Scaling
18 from sklearn.preprocessing import StandardScaler
19 sc = StandardScaler()
                                                                       best_parameters - Dictionary (3 elements)
20 X_train = sc.fit_transform(X_train)
21 X test = sc.transform(X test)
23 # Fitting Kernel SVM to the Training set
                                                                C
                                                                        int
                                                                               1 1
24 from sklearn.svm import SVC
                                                                        float
                                                                               1 0
                                                                gamma
25 classifier = SVC(kernel = 'rbf', random_state = 0)
26 classifier.fit(X_train, y_train)
                                                                kernel str
                                                                               1 rbf
28 # Predicting the Test set results
29 y pred = classifier.predict(X test)
30
                                                                                           Cancel
                                                                                                   OK
31 # Making the Confusion Matrix
32 from sklearn.metrics import confusion matrix
33 cm = confusion matrix(y test, y pred)
35 # Applying k-Fold Cross Validation
36 from sklearn.model selection import cross val score
37 accuracies = cross val score(estimator = classifier, X = X train, y = y train, cv = 10)
38 accuracies.mean()
39 accuracies.std()
40
41 # Applying Grid Search to find the best model and the best parameters
42 from sklearn.model_selection import GridSearchCV
43 parameters = [{'C': [1, 10, 100, 1000], 'kernel': ['linear']},
44 {'C': [1, 10, 100, 1000], 'kernel': ['rbf'], 'gamma': [0.5, 0.1, 0.01, 0.001, 0.0001]}]
45 grid_search = GridSearchCV(estimator = classifier,
                               param grid = parameters,
47
                               scoring = 'accuracy',
```

Grid Search

```
6 import pandas as pd
 8 # Importing the dataset
 9 dataset = pd.read_csv('Social_Network_Ads.csv')
10 X = dataset.iloc[:, [2, 3]].values
11 y = dataset.iloc[:, 4].values
13 # Splitting the dataset into the Training set and Test set
14 from sklearn.model selection import train test split
15 X train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
17 # Feature Scaling
18 from sklearn.preprocessing import StandardScaler
19 sc = StandardScaler()
20 X_train = sc.fit_transform(X_train)
21 X test = sc.transform(X test)
23 # Fitting Kernel SVM to the Training set
24 from sklearn.svm import SVC
25 classifier = SVC(kernel = 'rbf', random_state = 0)
26 classifier.fit(X_train, y_train)
28 # Predicting the Test set results
29 y pred = classifier.predict(X test)
31 # Making the Confusion Matrix
32 from sklearn.metrics import confusion matrix
33 cm = confusion_matrix(y_test, y_pred)
35 # Applying k-Fold Cross Validation
36 from sklearn.model selection import cross val score
37 accuracies = cross_val_score(estimator = classifier, X = X_train, y = y_train, cv = 10)
38 accuracies.mean()
39 accuracies.std()
41# Applying Grid Search to find the best model and the best parameters
42 from sklearn.model_selection import GridSearchCV
43 parameters = [{'C': [1, 10, 100, 1000], 'kernel': ['linear']},
44 {'C': [1, 10, 100, 1000], 'kernel': ['rbf'], 'gamma': [0.5, 0.1, 0.01, 0.001, 0.0001]}]
45 grid_search = GridSearchCV(estimator = classifier,
                               param grid = parameters,
47
                               scoring = 'accuracy',
48
                               cv = 10,
                               n \text{ jobs} = -1
50 grid search = grid search.fit(X train, y train)
51 best_accuracy = grid_search.best_score
52 best parameters = grid search.best params
```

```
8 # Importing the dataset
                                                                                                                 Name
9 dataset = pd.read csv('Social Network Ads.csv')
                                                                                                           х
0 X = dataset.iloc[:, [2, 3]].values
ly = dataset.iloc[:, 4].values
                                                                                                           X test
                                                                                                           X train
3 # Splitting the dataset into the Training set and Test set
                                                                                                           accuracies
4 from sklearn.model selection import train test split
5 X train, X test, y train, y test = train test split(X, y, test size = 0.25, random state = 0)
                                                                                                           best accuracy
7 # Feature Scaling
                                                                                                           best partmeters
8 from sklearn.preprocessing import StandardScaler
9 sc = StandardScaler()
0 X train = sc.fit transform(X train)
                                                                                                           dataset
1 X test = sc.transform(X test)
                                                                                                           parameters
3 # Fitting Kernel SVM to the Training set
4 from sklearn.svm import SVC
                                                                                                           y pred
5 classifier = SVC(kernel = 'rbf', random state = 0)
6 classifier.fit(X train, y train)
                                                                                                           y_test
                                                                                                           y train
8 # Predicting the Test set results
                                                                                           best_parameters - Dictionary (3 elements)
9 y pred = classifier.predict(X test)
1 # Making the Confusion Matrix
                                                                                                             Value
2 from sklearn.metrics import confusion matrix
                                                                                            int
                                                                                                   1 1
3 cm = confusion_matrix(y_test, y_pred)
                                                                                    gamma
                                                                                            float 1 0.7
5 # Applying k-Fold Cross Validation
                                                                                    kernel
                                                                                            str
                                                                                                   1 rbf
6 from sklearn.model selection import cross val score
7 accuracies = cross val score(estimator = classifier, X = X train, y = y train, cv
8 accuracies.mean()
9 accuracies.std()
                                                                                                             Cancel
1 # Applying Grid Search to find the best model and the best parameters
                                                                                                           In [7]: best pa
2 from sklearn.model selection import GridSearchCV
In [8]: from sk
                                                                                                              ...: paramet
5 grid search = GridSearchCV(estimator = classifier,
                                                                                                           ['linear']},
                            param grid = parameters,
                                                                                                              . . . :
7
                            scoring = 'accuracy',
                                                                                                           ['rbf'], 'gamma
8
                            cv = 10,
                                                                                                           0.91}1
                            n jobs = -1)
                                                                                                              ...: grid se
Ogrid search = grid search fit(X train, y train)
```

XGBoost

```
5 # Importing the libraries
6 import numpy as np
 7 import matplotlib.pyplot as plt
 8 import pandas as pd
10 # Importing the dataset
11 dataset = pd.read_csv('Churn_Modelling.csv')
12 X = dataset.iloc[:, 3:13].values
13 y = dataset.iloc[:, 13].values
14
15 # Encoding categorical data
16 from sklearn.preprocessing import LabelEncoder, OneHotEncoder
17 labelencoder X 1 = LabelEncoder()
18 X[:, 1] = labelencoder X 1.fit transform(X[:, 1])
19 labelencoder X 2 = LabelEncoder()
20 X[:, 2] = labelencoder X 2.fit transform(X[:, 2])
21 onehotencoder = OneHotEncoder(categorical features = [1])
22 X = onehotencoder.fit transform(X).toarray()
23X = X[:, 1:]
24
25 # Splitting the dataset into the Training set and Test set
26 from sklearn.model selection import train test split
27 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
29 # Fitting XGBoost to the Training set
30 from xgboost import XGBClassifier
31 classifier = XGBClassifier()
32 classifier.fit(X train, y train)
33
34 # Predicting the Test set results
35 y pred = classifier.predict(X test)
36
37 # Making the Confusion Matrix
38 from sklearn.metrics import confusion matrix
39 cm = confusion matrix(y test, y pred)
41 # Applying k-Fold Cross Validation
42 from sklearn.model selection import cross val score
43 accuracies = cross val score(estimator = classifier, X = X train, y = y train, cv = 10)
44 accuracies.mean()
45 accuracies.std()
```