**K-Nearest Neighbors**

**########## knn ############**

**Parameters: {'n\_neighbors': 3, 'p': 1}**

**Score: 0.7145833333333333**

Parameters: {'n\_neighbors': 9, 'p': 1}

Score: 0.70625

Parameters: {'n\_neighbors': 4, 'p': 1}

Score: 0.7041666666666667

* Train:
  + Mean model ACC: 70.63%

Chart

Description automatically generated

* Test:
  + Mean model ACC: 58.33%

Chart

Description automatically generated

**Support Vector Machine**

**########## svm ############**

Parameters: {'C': 1000, 'gamma': 0.01, 'kernel': 'rbf'}

Score: 0.7833333333333333

Parameters: {'C': 100, 'gamma': 0.1, 'kernel': 'rbf'}

Score: 0.78125

**Parameters: {'C': 1000, 'gamma': 0.1, 'kernel': 'rbf'}**

**Score: 0.7666666666666667**

* Train:
  + ﻿Mean model ACC: 76.67%

Chart, treemap chart

Description automatically generated

* Test:
  + Mean model ACC: 70%

Chart

Description automatically generated

**MultiLayer Perceptron**

**﻿########## ann ############**

**﻿**Parameters: {'activation': 'tanh', 'alpha': 1, 'hidden\_layer\_sizes': (100, 30), 'max\_iter': 10000, 'solver': 'lbfgs'}

Score: 0.8145247923193514

**Parameters: {'activation': 'tanh', 'alpha': 0.1, 'hidden\_layer\_sizes': (50, 30), 'max\_iter': 10000, 'solver': 'lbfgs'}**

**Score: 0.8104408033518077**

Parameters: {'activation': 'tanh', 'alpha': 1, 'hidden\_layer\_sizes': (100, 20), 'max\_iter': 10000, 'solver': 'lbfgs'}

Score: 0.8102621346575509

* Train:
  + Mean model ACC: ﻿78.96%

Chart, treemap chart

Description automatically generated

* Test:
  + ﻿Mean model ACC: ﻿75%

Chart

Description automatically generated

**Random Forest Model**

**﻿**﻿########## rfc ############

**Parameters: {'criterion': 'entropy', 'max\_depth': 9, 'max\_features': 'log2', 'min\_samples\_leaf': 3, 'min\_samples\_split': 12, 'n\_estimators': 100}**

**Score: 0.775**

Parameters: {'criterion': 'entropy', 'max\_depth': 9, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 100}

Score: 0.7749999999999999

Parameters: {'criterion': 'log\_loss', 'max\_depth': 8, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 12, 'n\_estimators': 200}

Score: 0.7729166666666667

* Train:
  + Mean model ACC: ﻿76.67%

Chart, treemap chart

Description automatically generated

* Test:
  + Mean model ACC: ﻿70%

Chart

Description automatically generated

**Decision Tree Classifier**

**########## dtc ############**

**Parameters: {'criterion': 'entropy', 'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 3, 'min\_samples\_split': 10}**

**Score: 0.7333333333333334**

Parameters: {'criterion': 'entropy', 'max\_depth': 9, 'max\_features': 'sqrt', 'min\_samples\_leaf': 5, 'min\_samples\_split': 10}

Score: 0.7270833333333334

Parameters: {'criterion': 'gini', 'max\_depth': 9, 'max\_features': 'sqrt', 'min\_samples\_leaf': 5, 'min\_samples\_split': 8}

Score: 0.7250000000000001

* Train:
  + Mean model ACC: ﻿70.21%

**Chart, treemap chart

Description automatically generated**

* Test:
  + Mean model ACC: ﻿66.67%

**Chart

Description automatically generated**

**Gaussian Naïve Bayes**

**########## gnb ############**

**Parameters: {'var\_smoothing': 0.1}**

**Score: 0.6208333333333333**

Parameters: {'var\_smoothing': 0.01}

Score: 0.6083333333333334

Parameters: {'var\_smoothing': 0.0001}

Score: 0.6020833333333333

* Train:
  + Mean model ACC: ﻿62.08%

**Chart

Description automatically generated**

* Test:
  + Mean model ACC: ﻿61.67%

**Chart

Description automatically generated**