## Classification

1. **Random Forest**

from sklearn.ensemble import RandomForestClassifier

classifier = RandomForestClassifier(criterion="entropy", n\_estimators=10, random\_state=0)

A screenshot of a graph

AI-generated content may be incorrect.

1. What is the Overall performance of the model?

Accuracy – 90% (0.90)

1. What is the correct performance of purchase?

Recall (1) – 88% (0.88)

1. What is the correct performance of purchase including failure?

Precision (1) – 86% (0.86)

1. What is the F1 score of non-purchased?

F1-score (0) – 92% (0.92)

1. How many samples are in class 1?

Support (1) - 49

1. How many samples are in class 0?

Support (0) – 85

1. What is the macro-average precision?

Macro Avg (Precision) – 89% (0.89)

1. What is the weighted average recall?

Weighted Avg (Recall) – 90% (0.90)

1. What is the overall test set in this model?

134 (85+49)

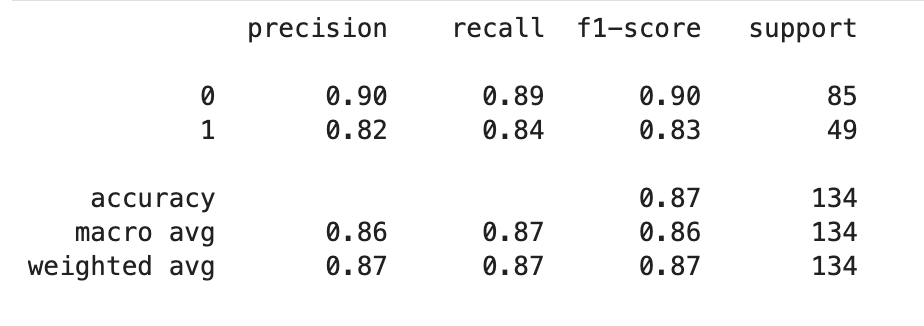
1. Is the dataset balanced based?

No (Because purchased is 49 and not purchased is 85)

1. **Decision Tree**

from sklearn.tree import DecisionTreeClassifier

classifier = DecisionTreeClassifier(criterion="entropy", random\_state=0, splitter='best')



1. What is the Overall performance of the model?

Accuracy – 87% (0.87)

1. What is the correct performance of purchase?

Recall (1) – 84% (0.84)

1. What is the correct performance of purchase including failure?

Precision (1) – 82% (0.82)

1. What is the F1 score of non-purchased?

F1-score (0) – 90% (0.90)

1. How many samples are in class 1?

Support (1) - 49

1. How many samples are in class 0?

Support (0) – 85

1. What is the macro-average precision?

Macro Avg (Precision) – 86% (0.86)

1. What is the weighted average recall?

Weighted Avg (Recall) – 87% (0.86)

1. What is the overall test set in this model?

134 (85+49)

1. Is the dataset balanced based?

No (Because purchased is 49 and not purchased is 85)

1. **Support Vector Classification - SVC**

from sklearn.svm import SVC

classifier = SVC(kernel="rbf", random\_state=0)

A number of numbers on a white background

AI-generated content may be incorrect.

1. What is the Overall performance of the model?

Accuracy – 78% (0.78)

1. What is the correct performance of purchase?

Recall (1) – 47% (0.47)

1. What is the correct performance of purchase including failure?

Precision (1) – 88% (0.88)

1. What is the F1 score of non-purchased?

F1-score (0) – 85% (0.85)

1. How many samples are in class 1?

Support (1) - 49

1. How many samples are in class 0?

Support (0) – 85

1. What is the macro-average precision?

Macro Avg (Precision) – 82% (0.82)

1. What is the weighted average recall?

Weighted Avg (Recall) – 78% (0.78)

1. What is the overall test set in this model?

134 (85+49)

1. Is the dataset balanced based?

No (Because purchased is 49 and not purchased is 85)

1. Logistics Regression (Sigmoid)

from sklearn.linear\_model import LogisticRegression

classifier = LogisticRegression(random\_state=0)

A number of numbers on a white background

AI-generated content may be incorrect.

1. What is the Overall performance of the model?

Accuracy – 89% (0.89)

1. What is the correct performance of purchase?

Recall (1) – 82% (0.82)

1. What is the correct performance of purchase including failure?

Precision (1) – 87% (0.87)

1. What is the F1 score of non-purchased?

F1-score (0) – 91% (0.91)

1. How many samples are in class 1?

Support (1) - 49

1. How many samples are in class 0?

Support (0) – 85

1. What is the macro-average precision?

Macro Avg (Precision) – 88% (0.88)

1. What is the weighted average recall?

Weighted Avg (Recall) – 89% (0.89)

1. What is the overall test set in this model?

134 (85+49)

1. Is the dataset balanced based?

No (Because purchased is 49 and not purchased is 85)

1. **K-Nearest Nighbor (KNN)**

from sklearn.neighbors import KNeighborsClassifier

classifier = KNeighborsClassifier(n\_neighbors=7, metric='minkowski', p=2)

A number of numbers on a white background

AI-generated content may be incorrect.

1. What is the Overall performance of the model?

Accuracy – 84% (0.84)

1. What is the correct performance of purchase?

Recall (1) – 67% (0.67)

1. What is the correct performance of purchase including failure?

Precision (1) – 85% (0.85)

1. What is the F1 score of non-purchased?

F1-score (0) – 88% (0.88)

1. How many samples are in class 1?

Support (1) - 49

1. How many samples are in class 0?

Support (0) – 85

1. What is the macro-average precision?

Macro Avg (Precision) – 84% (0.84)

1. What is the weighted average recall?

Weighted Avg (Recall) – 84% (0.84)

1. What is the overall test set in this model?

134 (85+49)

1. Is the dataset balanced based?

No (Because purchased is 49 and not purchased is 85)

1. **K-Nearest Nighbor (KNN)**
   1. **Gaussian Naive Bayes**

from sklearn.naive\_bayes import GaussianNB

classifier = GaussianNB()

A number of numbers in a row

AI-generated content may be incorrect.

* 1. **Multinomial Naive Bayes**

from sklearn.naive\_bayes import MultinomialNB

classifier = MultinomialNB()

A number of numbers in a row

AI-generated content may be incorrect.

* 1. **Complement Naive Bayes**

from sklearn.naive\_bayes import ComplementNB

classifier = ComplementNB()

A number of numbers in a row

AI-generated content may be incorrect.

* 1. **Bernoulli Naive Bayes**

from sklearn.naive\_bayes import BernoulliNB

classifier = BernoulliNB()

A number of numbers in a row

AI-generated content may be incorrect.

* 1. **Categorical Naive Bayes**

from sklearn.naive\_bayes import CategoricalNB

classifier = CategoricalNB()

A number of numbers in a row

AI-generated content may be incorrect.