Example output for holiday inn

Test 1 -> Logistic Regression -> Compare accuracy (% of correct predictions) vs precision (correct positive predictions out of total positive predictions) vs recall (correct positive predictions out of total actual positives

Shape, square

Description automatically generated

**TP -> True Positives** intersection of true predictions with true labels -> represent data points that have a True label which was correctly predicted by our algorithm or model

**FP -> False Positives** intersection of True predictions with False real labels -> represent data points that have a False real label but that has been predicted as True by the model

**FN -> False Negatives ->** intersection of False predictions with True real labels -> represent data points that have a True real label that has been incorrectly predicted as False by the model

**TN -> True Negatives ->** intersection between False predictions and False real labels -> represent data points that have a False real label that has been correctly classified as False by the model

0 = Negative, 1 = Positive -> matrix layout is FP(blue), TN(red), FN(purple), TP (red) -> meaning 1 negative review but got flag as positive + 6 negative reviews correctly flag as negative + 0 positive review but got flag as negative + 6 positive reviews correctly flag as positive

# logistic regression

model\_lr = LogisticRegression(C=5, penalty='l2', random\_state=0)

model\_lr.fit(X\_train, y\_train)

c\_matrix(X\_train, X\_test, y\_train, y\_test, model\_lr)

use L2 regulation technique by default -> ridge regression add squared magnitude of coefficient as penalty term to loss function ie. if lambda too big may add weight and lead to underfitting -> result can conclude that sentiment analysis result is 100% accurate using logistic regression

Test 2 -> K Nearest Neighbors

A picture containing text, monitor, screen

Description automatically generated

read confusion matrix in same direction -> 0 = Negative, 1 = Positive -> matrix layout is FP(turquoise), TN(red), FN(purple), TP (orange) -> meaning 2 negative reviews but got flag as positive + 6 negative reviews correctly flag as negative + 0 positive review but got flag as negative + 5 positive reviews correctly flag as positive

# KNN test

model\_knn = KNeighborsClassifier(n\_neighbors=3, metric='euclidean')

model\_knn.fit(X\_train, y\_train)

c\_matrix(X\_train, X\_test, y\_train, y\_test, model\_knn)

use default find 5 nearest neighbours and Euclidean distance (for l2 regression case following first test) -> result can conclude that sentiment analysis result is 85% accurate using KNN

Test 3 -> Random Forest Classifier

Test 4 -> Support Vector Classifier

Test5 -> Naive Bayes -> classifier used for classification with discrete features eg. word counts for text classification -> compare with Test 1 the results should be same

model\_naive = MultinomialNB()

model\_naive.fit(X\_train, y\_train)

c\_matrix(X\_train, X\_test, y\_train, y\_test, model\_naive)

examples for datafiniti

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

Chart

Description automatically generated with low confidence