Quiz semana 3

1. 0.99 C
2. (19+8)/(19+8+4+96)=0.212 In, (19+96)/(19+96+4+8)=0.905 C
3. 96/(96+8)=0.923 C
4. 96/(96+4)=0.960 C
5. O.7 In, 0.700 In, 0.640 C

from sklearn.metrics import precision\_recall\_curve

y\_scores\_m = m.fit(X\_train, y\_train).predict(X\_test)

precision, recall, thresholds = precision\_recall\_curve(y\_test, y\_scores\_m)

# print(precision)

# print(recall)

# print(thresholds)

closest\_zero = np.argmin(np.abs(thresholds))

closest\_zero\_p = precision[closest\_zero]

closest\_zero\_r = recall[closest\_zero]

plt.figure(figsize=(9,5))

plt.xlim([0.0, 1.01])

plt.ylim([0.0, 1.01])

plt.plot(precision, recall, label='Precision-Recall Curve')

plt.plot(closest\_zero\_p, closest\_zero\_r, 'o', markersize = 12, fillstyle = 'none', c='r', mew=3)

plt.xlabel('Precision', fontsize=16)

plt.ylabel('Recall', fontsize=16)

plt.grid(visible='True')

plt.show()

1. Model 1 ROC 1, Model 2 ROC 3, Model 3 ROC 2. C
2. Model 1 ROC 1, Model 2 ROC 2, Model 3 ROC 3. In

Model 1 ROC 1, Model 2 ROC 3, Model 3 ROC 2. In

Model 1 ROC 2, Model 2 ROC 3, Model 3 ROC 1. In

Model 1 ROC 3, Model 2 ROC 2, Model 3 ROC 1. In

Not enoguh info C

1. Si es micro: 0.744(micro) C, si es macro 0.805 (macro) C.

from sklearn.metrics import average\_precision\_score

y\_test\_predicted = m.fit(X\_train, y\_train).predict(X\_test)

average=precision\_score(y\_test, y\_test\_predicted, average='macro')

print(average)

1. The best 1.0, A model that always 0.0, the worst 0.0 In

The best 1.0, A model that always negative, the worst 0.0 In

The best 1.0, the worst 0.0 In

The best 1.0 In

All answers C

1. Accuracy In, Precision C
2. Recall C
3. Misclassifying frecuent C
4. -0.038 In, 0.038 In,

from sklearn.model\_selection import GridSearchCV

from sklearn.metrics import recall\_score

from sklearn.metrics import precision\_score

grid\_values = {'gamma': [0.01, 0.1, 1, 10], 'C': [0.01, 0.1, 1, 10]}

grid= GridSearchCV(m, param\_grid = grid\_values, scoring ='recall').fit(X\_test, y\_test)

print('Grid best parameter (max. accuracy): ', grid.best\_params\_)

y\_decision\_fn\_scores= grid.predict(X\_test)

recall = recall\_score(y\_test, y\_decision\_fn\_scores)

precision= precision\_score(y\_test, y\_decision\_fn\_scores)

ans = recall - precision

-0.038

1. 0.150 C

from sklearn.metrics import recall\_score

from sklearn.metrics import precision\_score

grid\_values = {'gamma': [0.01, 0.1, 1, 10], 'C': [0.01, 0.1, 1, 10]}

grid= GridSearchCV(m, param\_grid = grid\_values, scoring ='precision').fit(X\_test, y\_test)

print('Grid best parameter (max. accuracy): ', grid.best\_params\_)

y\_decision\_fn\_scores= grid.predict(X\_test)

recall = recall\_score(y\_test, y\_decision\_fn\_scores)

precision= precision\_score(y\_test, y\_decision\_fn\_scores)

ans = precision - recall

0.150