Tucil 1 IF3270 Machine Learning dengan Python

13519153 Maximillian Lukman - 13519162 Irvin Andryan Pratomo

In [98]:
```
from sklearn import datasets, tree, model_selection, preprocessing, metrics, svm, cluster, linear_model, neura
```

In [100]:
```
from id3 import Id3Estimator, export_text
```

In [101]:
```
import pandas as pd
```

Decision Tree Classifier

Id3Estimator

```
worst radius >16.80
|   mean texture <=14.99
|   |   mean compactness <=0.13: 1 (5)
|   |   mean compactness >0.13: 0 (2)
|   mean texture >14.99
|   |   worst concavity <=0.20
|   |   |   mean compactness <=0.07: 0 (3)
|   |   |   mean compactness >0.07: 1 (2)
|   |   worst concavity >0.20: 0 (144)
```

In [120]:
```
prediction_cancer_Id3 = estimator_cancer.predict(X_test_cancer)
print("Id3Estimator: Cancer")
print("Accuracy:", metrics.accuracy_score(y_test_cancer, prediction_cancer_Id3))
print("F1 Macro avg:", metrics.f1_score(y_test_cancer, prediction_cancer_Id3, average="macro"))
print("F1 Weighted avg:", metrics.f1_score(y_test_cancer, prediction_cancer_Id3, average="weighted"))
```
```
Id3Estimator: Cancer
Accuracy: 0.9122807017543859
F1 Macro avg: 0.8952205882352942
F1 Weighted avg: 0.9122807017543859
```

K-Means

In [121]:
```
kmeans = cluster.KMeans(n_clusters=2)
```

In [122]:
```
kmeans_tennis = kmeans.fit(X_train_tennis,y_train_tennis)
```

In [123]:
```
kmeans_tennis_centroid = pd.DataFrame(kmeans_tennis.cluster_centers_.transpose())
kmeans_tennis_centroid.index = tennis.columns[:-1].tolist()
kmeans_tennis_centroid.columns = ["Centroid 1", "Centroid 2"]
print(kmeans_tennis_centroid)
```
```
          Centroid 1  Centroid 2
outlook          1.2    0.666667
temp             1.8    0.333333
humidity         0.2    0.833333
wind             0.2    0.666667
```

In [124]:
```
prediction_tennis_kmeans = kmeans_tennis.predict(X_test_tennis)
print("K-Means: Tennis")
print("Accuracy:", metrics.accuracy_score(y_test_tennis, prediction_tennis_kmeans))
print("F1 Macro avg:", metrics.f1_score(y_test_tennis, prediction_tennis_kmeans, average="macro"))
print("F1 Weighted avg:", metrics.f1_score(y_test_tennis, prediction_tennis_kmeans, average="weighted"))
```
```
K-Means: Tennis
Accuracy: 0.6666666666666666
F1 Macro avg: 0.4
F1 Weighted avg: 0.5333333333333333
```

In [125]:
```
kmeans_cancer = kmeans.fit(X_train_cancer,y_train_cancer)
```

In [126]:
```
kmeans_cancer_centroid = pd.DataFrame(kmeans_cancer.cluster_centers_.transpose())
kmeans_cancer_centroid.index = cancer['feature_names'].tolist()
kmeans_cancer_centroid.columns = ["Centroid 1", "Centroid 2"]
print(kmeans_cancer_centroid)
```
```
                          Centroid 1   Centroid 2
mean radius                12.630837    19.587429
mean texture               18.618429    21.724476
mean perimeter             81.683886   129.484762
mean area                 501.985429  1206.572381
mean smoothness             0.094825     0.101158
mean compactness            0.092810     0.146516
mean concavity              0.064946     0.175317
mean concave points         0.034320     0.101167
mean symmetry               0.178605     0.190745
mean fractal dimension      0.063443     0.060226
radius error                0.302593     0.725070
texture error               1.196623     1.239931
perimeter error             2.166278     5.121648
area error                 24.009317    92.785714
smoothness error            0.007092     0.006614
compactness error           0.023842     0.031844
concavity error             0.029107     0.042084
concave points error        0.010629     0.015767
symmetry error              0.020467     0.020688
fractal dimension error     0.003699     0.003944
worst radius               14.162960    23.785714
worst texture              24.897171    28.994000
worst perimeter            92.894286   158.588571
worst area                630.333714  1757.771429
worst smoothness            0.130538     0.139894
worst compactness           0.230533     0.350978
worst concavity             0.229018     0.441687
worst concave points        0.093537     0.192800
worst symmetry              0.286359     0.311059
worst fractal dimension     0.083832     0.085536
```

In [127]:
```
prediction_cancer_kmeans = kmeans_cancer.predict(X_test_cancer)
print("K-Means: Cancer")
print("Accuracy:", metrics.accuracy_score(y_test_cancer, prediction_cancer_kmeans))
print("F1 Macro avg:", metrics.f1_score(y_test_cancer, prediction_cancer_kmeans, average="macro"))
print("F1 Weighted avg:", metrics.f1_score(y_test_cancer, prediction_cancer_kmeans, average="weighted"))
```
```
K-Means: Cancer
Accuracy: 0.10526315789473684
F1 Macro avg: 0.09832506203473944
F1 Weighted avg: 0.06640982107875146
```

Logistic Regression

In [128]:
```
lr = linear_model.LogisticRegression(class_weight='balanced', max_iter=2500, random_state=0)
```

In [129]:
```
lr_tennis = lr.fit(X_train_tennis,y_train_tennis)
```

In [130]:
```
prediction_tennis_lr = lr_tennis.predict(X_test_tennis)
print("Logistic Regression: Tennis")
print("Accuracy:", metrics.accuracy_score(y_test_tennis, prediction_tennis_lr))
print("F1 Macro avg:", metrics.f1_score(y_test_tennis, prediction_tennis_lr, average="macro"))
print("F1 Weighted avg:", metrics.f1_score(y_test_tennis, prediction_tennis_lr, average="weighted"))
```
```
Logistic Regression: Tennis
Accuracy: 0.6666666666666666
F1 Macro avg: 0.6666666666666666
F1 Weighted avg: 0.6666666666666666
```

In [131]:
```
lr_tennis_coefficient = pd.DataFrame(lr_tennis.coef_.transpose())
lr_tennis_coefficient.index = tennis.columns[:-1].tolist()
lr_tennis_coefficient.columns = ["Coefficient"]
print("Play Tennis")
print("")
print(lr_tennis_coefficient)
```
```
Play Tennis

          Coefficient
outlook     -0.594406
temp         0.197587
humidity     0.511366
wind         1.033502
```

In [132]:
```
lr_cancer = lr.fit(X_train_cancer,y_train_cancer)
```

In [133]:
```
prediction_cancer_lr = lr_cancer.predict(X_test_cancer)
print("Logistic Regression: Cancer")
print("Accuracy:", metrics.accuracy_score(y_test_cancer, prediction_cancer_lr))
print("F1 Macro avg:", metrics.f1_score(y_test_cancer, prediction_cancer_lr, average="macro"))
print("F1 Weighted avg:", metrics.f1_score(y_test_cancer, prediction_cancer_lr, average="weighted"))
```
```
Logistic Regression: Cancer
Accuracy: 0.956140350877193
F1 Macro avg: 0.9480448455017774
F1 Weighted avg: 0.9563202509966466
```

In [134]:
```
lr_cancer_coefficient = pd.DataFrame(lr_cancer.coef_.transpose())
lr_cancer_coefficient.index = cancer.feature_names.tolist()
lr_cancer_coefficient.columns = ["Coefficient"]
print("Breast Cancer")
print("")
print(lr_cancer_coefficient)
```
```
Breast Cancer

                         Coefficient
mean radius                 0.956030
mean texture                0.219475
mean perimeter             -0.396222
mean area                   0.030067
mean smoothness            -0.159605
mean compactness           -0.238834
mean concavity             -0.453349
mean concave points        -0.250831
mean symmetry              -0.272578
mean fractal dimension     -0.035377
radius error               -0.018472
texture error               1.699444
perimeter error             0.163954
area error                 -0.115496
smoothness error           -0.022168
compactness error           0.057191
concavity error            -0.028179
concave points error       -0.031378
symmetry error             -0.044235
fractal dimension error     0.013249
worst radius                0.228848
worst texture              -0.516900
worst perimeter            -0.084322
worst area                 -0.015815
worst smoothness           -0.321128
worst compactness          -0.805459
worst concavity            -1.333088
worst concave points       -0.552800
worst symmetry             -0.782240
worst fractal dimension    -0.113955
```

Neural Network

In [135]:
```
neural = neural_network.MLPClassifier(max_iter=1000)
```

In [136]:
```
neural_tennis = neural.fit(X_train_tennis,y_train_tennis)
```

In [137]:
```
prediction_tennis_neural = neural_tennis.predict(X_test_tennis)
print("Neural Network: Tennis")
print("Accuracy:", metrics.accuracy_score(y_test_tennis, prediction_tennis_neural))
print("F1 Macro avg:", metrics.f1_score(y_test_tennis, prediction_tennis_neural, average="macro"))
print("F1 Weighted:", metrics.f1_score(y_test_tennis, prediction_tennis_neural, average="weighted"))
```
```
Neural Network: Tennis
Accuracy: 0.3333333333333333
F1 Macro avg: 0.25
F1 Weighted: 0.16666666666666666
```

In [138]:
```
neural_tennis_coefficient = neural_tennis.coefs_
print("Play Tennis Coefficient")
print("")
print(neural_tennis_coefficient)
```
```
Play Tennis Coefficient

[array([[ 2.97168634e-01, -2.75355048e-01, -3.72041281e-01,
        -9.06265515e-02, -2.21376682e-01, -1.20596133e-01,
         1.88444635e-01, -3.54340810e-01,  6.61429659e-15,
        -5.28094546e-02, -3.81503808e-01,  4.49444827e-01,
        -1.95213540e-02,  1.37185919e-01,  2.20672159e-12,
         5.86934673e-17, -1.02676444e-01,  2.66385363e-01,
        -4.31563274e-01,  2.10394437e-01,  1.80523357e-01,
        -3.11013355e-01, -4.47417100e-01, -2.03996360e-01,
        -2.81119336e-01,  3.11124510e-01,  3.41082278e-01,
        -1.44133866e-01,  1.57995447e-02,  2.80851476e-01,
        -7.70803262e-19, -3.49291742e-01,  2.74176017e-01,
         5.13885296e-01,  2.69984061e-01, -1.26441045e-02,
         3.11645708e-01, -5.18790616e-01,  1.19110390e-01,
         1.57711166e-01,  3.39626225e-01, -3.42221839e-01,
        -2.70517511e-01, -1.34756345e-01, -1.38893903e-01,
        -9.10015162e-02, -5.04658360e-02,  2.71309403e-06,
         5.53541278e-02,  4.32160084e-02,  5.97821284e-03,
         3.10139133e-01,  1.77522511e-01, -1.88450399e-01,
        -8.23213928e-02, -2.75422024e-04,  3.29975018e-01,
        -8.53137598e-02, -5.84935439e-04,  4.13871186e-01,
         4.17478290e-03,  2.50034010e-01,  9.47225182e-02,
         5.43758125e-04,  3.00828106e-01,  3.47799291e-01,
         2.51413517e-01,  3.60288840e-01, -8.11275504e-03,
        -5.80145139e-02,  2.43757943e-01,  4.20669505e-01,
        -5.23198606e-01,  1.65709255e-16,  4.71571963e-02,
         2.18371071e-01, -1.10982144e-01, -3.79039707e-01,
         3.33609704e-01,  7.51543931e-01,  1.99877267e-01,
         2.41442027e-01,  2.85976161e-02,  1.45188176e-01,
        -1.46621133e-01,  2.53797111e-01,  2.31103443e-01,
        -1.52540022e-01,  4.29239649e-01,  2.37039071e-01,
         2.92877413e-01,  1.19418894e-01,  4.46748704e-01,
        -1.41400273e-02, -3.24052626e-02, -1.75981948e-01,
        -3.80060301e-01,  3.30143100e-01,  2.43518785e-01,
        -3.71636045e-01],
       [-3.55057601e-01,  3.04441218e-01,  1.86033415e-01,
        -2.12290526e-02,  3.54578979e-01,  1.12371168e-01,
        -4.95445190e-02,  2.28912050e-01,  4.40017382e-16,
         1.69293884e-01,  4.05281997e-01,  1.70855796e-16,
        -7.48975676e-03,  2.33635333e-01,  1.71578431e-16,
         5.52797814e-01,  1.77553079e-01,  1.69808827e-01,
         1.35028855e-01, -1.64824832e-01,  6.01006819e-02,
         1.27039676e-02,  1.56222679e-01, -8.58629634e-03,
         2.37450603e-01, -3.81103057e-01,  2.34679839e-02,
        -4.41499492e-02,  2.69141945e-01,  1.04668450e-01,
        -3.22889247e-04,  4.35446611e-02,  4.41002819e-01,
        -9.45110527e-02,  8.83998655e-02, -7.94976426e-02,
        -1.43008272e-01,  7.36782198e-02,  2.46776782e-01,
        -8.04386764e-02, -1.41309198e-01,  1.84705924e-01,
         4.47325379e-02,  3.33229559e-01,  3.92197819e-01,
         4.56076297e-01,  3.84591014e-01, -1.19938554e-03,
         2.51007365e-03, -3.02863666e-01,  4.96120440e-03,
         1.09365733e-01,  6.38839315e-02, -9.68048500e-02,
         2.29759996e-02, -4.00486602e-03, -2.09403933e-02,
         1.87766091e-01, -7.85692057e-07,  1.05906781e-01,
        -3.65658542e-01,  1.49555961e-01,  4.51832306e-01,
         2.88869281e-01, -1.18553705e-01, -2.31479963e-03,
        -1.02059690e-01,  9.03438806e-02,  4.05882577e-02,
        -9.05062711e-03, -5.22651366e-01, -1.97637233e-01,
         1.19510541e-01,  6.50353342e-01, -1.81192578e-01,
        -1.85560596e-01, -3.88007784e-03,  2.25607186e-01,
        -4.02093091e-01,  5.47255681e-02,  4.62881222e-02,
        -3.30827880e-01,  2.56777345e-01,  1.25143059e-01,
         1.69738631e-01, -1.78548719e-01,  3.28682513e-01,
         3.42746302e-01, -2.23628333e-01, -5.01104500e-02,
         7.16197515e-02, -1.14211579e-01,  1.98530423e-02,
         2.55054800e-01, -1.95189781e-01,  1.83997526e-01,
         5.55597428e-02, -4.10600218e-01, -4.24161850e-01,
         2.85933558e-01],
       [-6.60621367e-02,  3.73272579e-01,  3.67870300e-01,
         5.90946749e-01,  3.48592753e-01,  4.69557408e-01,
         1.17227478e-01,  1.58406291e-01,  6.91074651e-13,
         3.44605324e-01,  1.80105684e-01, -2.63656956e-01,
        -8.28153799e-02,  2.40757245e-01, -1.38558745e-03,
         3.27817909e-13,  1.64248464e-01, -3.02185780e-01,
         2.69626076e-01,  1.65812768e-01,  4.63012636e-01,
         2.57080826e-02,  3.12522887e-01, -1.69637574e-02,
         4.08757192e-01, -1.00346030e-01, -1.46973662e-01,
        -8.89976274e-02,  1.09787356e-01,  1.10555329e-01,
        -1.62431321e-04,  8.76757216e-02,  2.10553499e-01,
        -5.57369593e-01, -1.81572290e-01, -4.64056835e-02,
         4.20257582e-01,  1.47541244e-01, -4.66764317e-01,
        -1.89590104e-01, -4.39193312e-02,  2.75643179e-02,
         8.87935492e-02,  2.89317845e-01,  3.96516199e-01,
         6.79301893e-01,  1.77348622e-02, -1.09486448e-07,
         4.42714901e-01,  3.49231049e-01,  3.11850033e-02,
        -4.04166069e-02, -2.23436553e-01,  2.37316329e-02,
         2.70599696e-01, -3.74815428e-01,  5.81733410e-01,
         1.18706058e-01,  2.04039458e-01, -1.30832825e-01,
         4.88257052e-01,  1.27851644e-01,  1.19281812e-01,
         6.35670267e-02,  6.00560732e-02, -3.21519257e-01,
         3.67497656e-02, -4.58759577e-01, -1.40057238e-01,
         1.77920557e-01,  1.16733505e-01, -7.28836183e-01,
         2.37224645e-01, -4.09604057e-01, -2.04407581e-01,
         4.00282070e-02,  3.56745920e-01,  6.51021449e-01,
         2.31551124e-01,  3.17868984e-01,  1.91701528e-01,
         3.28848239e-02,  4.61601137e-01,  3.03049188e-01,
         3.39850831e-01, -5.86401761e-01, -4.76269434e-03,
         6.99386436e-01, -4.45377475e-01, -4.17553170e-01,
        -4.82456403e-02,  3.85040583e-01,  6.81855989e-02,
        -5.97878479e-01,  2.47741064e-01,  2.09080689e-01,
         1.10937853e-01, -1.54471116e-01, -2.56031997e-02,
         3.31675877e-01],
       [ 4.01516957e-01, -3.62401582e-02,  1.83723870e-01,
        -4.56335781e-01,  3.86577965e-01,  4.80786070e-02,
         7.29634938e-02,  2.20994255e-01,  1.61604256e-15,
         3.78231525e-01, -8.99791618e-03, -5.39606797e-01,
         1.43285659e-01,  3.30134285e-01,  9.06143251e-16,
        -1.35063723e-17,  3.72458287e-02, -2.67245480e-01,
         2.87134181e-01,  3.96379947e-01,  3.26917798e-01,
         3.43877306e-01,  3.07841011e-01,  4.02227558e-02,
         4.10713613e-01,  4.50015339e-01, -2.25850793e-01,
         5.43422080e-01,  4.41172448e-01,  4.50739282e-01,
         3.43256438e-17,  5.39561281e-02, -6.01332292e-01,
        -3.07167658e-01, -2.11148921e-01,  2.23653365e-02,
         2.04596333e-01,  2.35454820e-01, -2.52260557e-01,
         4.74840453e-01,  3.13703763e-01, -1.82129919e-01,
         9.18067249e-02,  3.40748619e-01,  1.09410990e-02,
         3.37360352e-01,  1.28159658e-01, -1.07545018e-01,
        -1.36903087e-01, -3.70228901e-01, -2.73967180e-01,
        -2.85897071e-01, -2.20689508e-02,  1.36132047e-01,
         3.69127585e-01, -4.88498118e-03,  1.23172210e-01,
         4.54629276e-02, -5.59607498e-16, -2.62268164e-01,
        -6.10646346e-01, -1.42223272e-01, -3.86569892e-01,
         2.32675273e-01,  4.38879302e-01, -3.94737296e-02,
         9.18744245e-02, -2.49971511e-01, -1.45967475e-01,
         3.64129704e-01, -6.28153969e-01,  3.07111609e-02,
         2.10839577e-01, -4.02547153e-01, -3.05177155e-01,
         4.86768466e-01,  3.05603027e-01,  3.57230763e-01,
        -1.96887078e-01, -6.65097326e-02,  1.11174431e-01,
        -3.06098230e-01,  9.93222440e-02, -2.38624226e-01,
         2.15278259e-01, -5.05930297e-01, -1.92694417e-02,
         1.43308498e-01, -4.38202990e-01,  4.74210300e-01,
         1.17493140e-02,  4.36124746e-01,  5.58042108e-01,
        -2.63600054e-01, -1.24479329e-02,  1.74983944e-01,
         2.39902356e-01, -1.48367980e-01, -3.88360133e-01,
         3.05081867e-01]]), array([[ 2.76389519e-01],
       [ 6.07843507e-01],
       [ 6.00426106e-16],
       [ 2.66574017e-01],
       [ 2.62473086e-01],
       [ 2.81850554e-01],
       [ 6.54327394e-02],
       [ 8.12970604e-01],
       [-1.25079269e-06],
       [ 4.56202634e-01],
       [ 2.76596595e-01],
       [-3.57482204e-01],
       [-1.62400963e-02],
       [ 5.57932748e-01],
       [-2.00486021e-03],
       [-1.83722180e-12],
       [ 4.42167169e-01],
       [-3.29046641e-01],
       [ 7.93532851e-01],
       [ 1.23137505e-01],
       [ 3.93343460e-01],
       [ 4.34479625e-01],
       [ 5.31863962e-01],
       [ 4.13317548e-01],
       [ 2.55747796e-01],
       [ 4.02248521e-01],
       [-5.68352120e-01],
       [ 5.79109331e-01],
       [ 4.01669458e-01],
       [ 4.93732390e-01],
       [-4.82040821e-16],
       [ 5.56265509e-01],
       [-5.80308756e-01],
       [-3.71609647e-01],
       [-5.08110212e-01],
       [-1.44175270e-01],
       [ 6.72855465e-01],
       [ 4.72891959e-01],
       [-6.11132704e-01],
       [ 5.69224432e-02],
       [-4.98312006e-01],
       [-2.93200346e-02],
       [-1.26471034e-01],
       [ 3.77016628e-01],
       [ 3.98564496e-01],
       [ 4.23424816e-01],
       [-2.15759805e-01],
       [-2.03893808e-03],
       [-4.45263988e-01],
       [-5.61266428e-01],
       [-2.98789888e-01],
       [-5.07166361e-01],
       [-8.42945108e-02],
       [-1.13045192e-01],
       [ 4.34398315e-01],
       [-6.84508456e-04],
       [ 7.51628166e-01],
       [ 5.05937849e-01],
       [-1.70958519e-02],
       [-4.64672912e-01],
       [-4.87293899e-01],
       [-2.29488757e-01],
       [-4.92455018e-01],
       [ 3.52605643e-01],
       [ 3.41836695e-01],
       [-4.49092793e-01],
       [-5.66171353e-02],
       [-4.50723508e-01],
       [ 7.07912849e-02],
       [ 3.60730272e-01],
       [-4.49775421e-01],
       [-4.07771998e-01],
       [ 6.38759475e-01],
       [-3.24467962e-04],
       [-4.05875879e-01],
       [ 5.97448930e-01],
       [ 3.50172572e-01],
       [ 5.98855834e-01],
       [-2.82812682e-01],
       [ 1.70577472e-01],
       [ 4.02802117e-01],
       [-7.06160384e-01],
       [ 4.05759751e-01],
       [ 7.07815310e-01],
       [ 5.59030802e-01],
       [-4.24147940e-01],
       [-3.26476559e-01],
       [ 2.50576707e-01],
       [-4.46338956e-01],
       [-7.33895485e-01],
       [-1.94376256e-01],
       [ 3.19262447e-01],
       [ 4.73907950e-01],
       [-2.56763088e-01],
       [-3.06131256e-01],
       [ 3.77109415e-01],
       [ 7.69463183e-01],
       [-5.45350463e-01],
       [-3.97744594e-01],
       [ 5.74659522e-01]])]
```

In [139]:
```
neural_cancer = neural.fit(X_train_cancer,y_train_cancer)
```

In [140]:
```
prediction_cancer_neural = neural_cancer.predict(X_test_cancer)
print("Neural Network: Cancer")
print("Accuracy:", metrics.accuracy_score(y_test_cancer, prediction_cancer_neural))
print("F1 Macro avg:", metrics.f1_score(y_test_cancer, prediction_cancer_neural, average="macro"))
print("F1 Weighted avg:", metrics.f1_score(y_test_cancer, prediction_cancer_neural, average="weighted"))
```
```
Neural Network: Cancer
Accuracy: 0.956140350877193
F1 Macro avg: 0.9480448455017774
F1 Weighted avg: 0.9563202509966466
```

In [141]:
```
neural_cancer_coefficient = neural_cancer.coefs_
print("Breast Cancer Coefficient")
print("")
print(neural_tennis_coefficient)
```

```
Breast Cancer Coefficient

[array([[ 2.97168634e-01, -2.75355044e-01, -3.72041281e-01,
         -9.06265515e-02, -2.21376682e-01, -1.20596133e-01,
          1.88444635e-01, -3.54340810e-01,  6.61429659e-15,
         -5.28094546e-02, -3.81503808e-01,  4.49444827e-01,
         -1.95213540e-02,  1.37185919e-01,  1.20672159e-12,
          5.86934673e-17, -1.02676464e-01,  2.66385365e-01,
         -4.51563274e-01,  2.10394437e-01, -1.80523375e-01,
         -3.11013355e-01, -4.47417100e-01, -2.03966360e-01,
         -2.81119334e-01,  3.11124510e-01,  3.41082278e-01,
         -1.44133846e-01,  1.57995447e-01,  2.80851474e-01,
         -7.71803262e-15, -2.49291742e-01,  2.74176017e-01,
          5.13885296e-01,  2.69944061e-01, -1.26441045e-02,
          3.11645708e-01, -5.18790618e-01,  1.19110390e-01,
          1.57711164e-01,  3.39626225e-01, -7.42221896e-03,
         -2.71517511e-01, -1.34756345e-01, -3.38879919e-01,
         -9.10015162e-02, -5.04658360e-02,  2.71309403e-06,
          5.53541278e-02,  4.32160084e-02,  5.97821284e-03,
          3.10139133e-01,  1.77522511e-01, -1.88450394e-01,
         -8.23213928e-02, -2.75422024e-04,  3.29975018e-01,
         -8.53137594e-02, -5.84935499e-04,  4.13871189e-01,
          4.17478290e-03,  2.50034010e-01,  9.47225182e-02,
          5.43758215e-04,  3.00828106e-01,  3.47799091e-01,
          2.51413517e-01,  3.60284840e-01, -8.11275504e-03,
         -5.80145139e-02,  2.43757943e-01,  4.20669505e-01,
         -5.23198606e-01,  1.65709265e-16,  4.71571960e-01,
          2.18371071e-01, -1.10982144e-01, -5.79039702e-01,
          3.33609704e-01,  7.51543931e-02,  1.99877240e-01,
          2.41442027e-01,  2.85976161e-02,  3.45188179e-01,
         -3.66621133e-01,  2.53797111e-01,  2.31103643e-01,
         -1.52540022e-01,  4.29239649e-01,  2.37039071e-01,
          2.92877413e-01,  2.19416894e-01,  4.46748704e-01,
         -1.41400273e-02, -3.24052626e-02, -2.75981948e-01,
         -3.80065051e-01,  3.30143100e-01,  2.43518785e-01,
         -9.71636045e-02],
        [-3.55057601e-01,  3.04641218e-01,  1.86633415e-01,
         -2.12290526e-02,  3.54578979e-01,  1.12371168e-01,
         -9.65445190e-02,  2.28912950e-01, -9.40017382e-14,
          1.69293884e-01,  4.05281997e-01,  1.70855796e-01,
         -7.48975678e-02,  2.33635393e-01, -1.71578431e-16,
          5.52797814e-11,  1.77553079e-01,  1.49808827e-01,
          1.35028853e-01, -1.64824832e-01,  6.01006619e-01,
          1.27039678e-02,  1.56222679e-01, -8.58629634e-03,
          6.27450603e-01, -3.81103053e-01,  2.34679830e-02,
         -4.41499492e-02,  2.69141945e-01,  1.04668450e-01,
         -3.22889247e-04,  4.35446611e-02, -4.41002819e-01,
         -9.45110527e-02,  6.83996655e-02, -7.94976426e-02,
         -1.43006272e-01,  7.36782198e-02,  2.46776782e-01,
         -8.04386764e-02, -1.41309198e-01,  1.84705924e-01,
          4.47325379e-02,  3.33229599e-01,  3.92197819e-01,
          4.56076297e-01,  3.84591014e-02, -1.19938554e-03,
          2.51007365e-01, -3.02863666e-01,  4.96120440e-02,
          1.09385733e-01,  6.38839315e-02, -9.68048501e-02,
          2.29793996e-01, -4.00486662e-02, -2.09410933e-02,
          1.87766091e-01, -7.85692057e-07,  1.05906781e-01,
         -3.65658542e-01, -3.49555961e-01, -4.51832309e-01,
          2.88869281e-01, -1.18553705e-01, -2.31479963e-03,
         -1.02059698e-01,  9.03435806e-02, -4.05882577e-02,
         -9.05062711e-03, -5.22651366e-01, -1.97637233e-01,
          1.18510541e-01,  6.50353342e-10, -1.81192578e-01,
         -1.45560596e-01, -3.88007784e-03,  2.25607280e-01,
         -4.62093091e-01,  5.47255681e-02,  9.62841222e-02,
         -3.30827880e-01,  2.56777345e-01,  1.25143059e-01,
          1.69738631e-01, -1.78546719e-01, -3.28682513e-01,
          3.42744302e-01, -2.23628333e-01, -5.01104500e-02,
          7.16197915e-02, -1.14211759e-01,  1.98530423e-02,
          2.55054900e-01, -1.95189781e-01,  3.83997520e-01,
          5.55597624e-02, -4.10660218e-01, -4.24141865e-01,
          2.85933558e-01],
        [-6.60621367e-02,  3.73272579e-01,  3.67870300e-01,
          5.90946749e-01,  3.48592753e-01,  4.69557404e-01,
          1.17227478e-01,  4.58406291e-01, -6.91074651e-13,
          3.44605324e-01,  1.80105684e-01, -2.63656958e-01,
         -8.28153799e-02,  2.40757245e-01, -1.38578858e-03,
          3.27817909e-13,  1.64248466e-01, -3.02185780e-01,
          2.69626076e-01,  1.65812768e-01,  6.63012638e-01,
          2.57080826e-02,  3.12522867e-01, -1.69637576e-02,
          4.08757192e-01, -1.00346030e-01, -1.46973662e-01,
         -8.89876274e-02,  1.09787356e-01,  2.10555325e-01,
         -1.62431321e-04,  8.76757216e-02,  1.20533498e-01,
         -5.57369593e-01, -1.81572290e-01, -6.64056831e-02,
          4.20257582e-01,  1.47141244e-01, -6.66764319e-01,
         -1.89590104e-01, -4.39193312e-01,  2.75843178e-02,
          8.87935492e-02,  2.89317845e-01,  3.96516197e-01,
          6.79301893e-01,  1.77348422e-02, -1.09486644e-07,
         -6.42714901e-01,  3.49231049e-01, -3.11850333e-02,
         -4.04166069e-01, -3.23443653e-02,  3.37316329e-02,
          2.70599694e-01, -3.74815428e-02,  5.81733410e-01,
          3.18706058e-01,  2.04039458e-01, -1.30832825e-01,
          4.86125702e-01,  1.17831644e-01,  3.19281612e-01,
          4.55567067e-01,  6.02060732e-01, -3.21519252e-01,
          3.67497656e-02, -4.58759937e-01, -1.40057238e-01,
          2.77920557e-01,  1.16735052e-01, -5.28834168e-01,
          2.37224645e-01, -4.09604057e-11, -2.04607581e-01,
         -4.00282070e-02,  3.56745990e-01,  4.51021244e-01,
          2.31151124e-01,  3.17868984e-01,  1.91701526e-01,
          3.28868229e-02,  4.61601137e-01,  3.03049188e-01,
          3.39850831e-01, -5.86401761e-01, -4.76269243e-03,
          6.99386438e-01, -4.45377475e-01, -4.17953713e-01,
         -4.82456403e-02,  3.85040583e-01,  6.81855989e-02,
         -5.97878479e-01,  2.47741064e-01,  2.09090689e-01,
          1.10937953e-01, -1.54471116e-01, -2.56031597e-02,
          3.31675877e-01],
        [ 4.01516957e-01, -3.62401582e-02,  1.83723478e-01,
          3.96012322e-01,  3.86577965e-01,  4.80784070e-01,
          7.29634934e-02,  3.20994255e-01,  1.61604256e-15,
          3.78231525e-01, -8.99791618e-03, -5.39606797e-01,
          1.43285659e-01,  3.30134285e-01,  9.06143251e-14,
         -1.33043725e-07,  3.72458287e-01, -2.67245480e-01,
          2.87134181e-01,  3.98379947e-03,  3.69971798e-01,
          3.43877306e-01,  3.07841011e-01,  4.02227538e-01,
          4.10713413e-01,  4.50015395e-01, -2.25850793e-01,
          5.43422080e-01,  4.41172448e-01,  4.50739285e-01,
          3.43256434e-17,  5.39561281e-01, -6.01332292e-01,
         -5.07167658e-01, -2.11148921e-01,  2.23853345e-01,
          2.04596333e-01,  2.35454820e-01, -2.52260557e-01,
          4.74840453e-02, -3.13703763e-01, -1.39297187e-01,
          9.18067249e-02,  3.40748619e-01,  1.09610990e-02,
          3.37360552e-01,  1.28159658e-01, -1.07545018e-03,
         -1.36903087e-01, -3.70228901e-01, -2.73967180e-01,
         -2.85897071e-01, -2.20689508e-02,  1.36132047e-01,
          3.69127585e-01, -4.88498119e-03,  1.23172210e-01,
          4.54629278e-01, -5.59607498e-16, -3.62268166e-01,
         -6.10646346e-01, -1.42223272e-01, -3.86569892e-01,
          2.32675273e-01,  4.38679302e-01, -5.94737298e-01,
          9.18744245e-02, -2.49971511e-01, -1.45967475e-01,
          3.64129706e-01, -6.28153992e-01, -3.07111609e-01,
          2.10839577e-01, -4.02547153e-10, -5.05177153e-01,
          4.86706446e-01,  3.05631027e-01,  3.57230763e-01,
         -1.96887070e-01, -6.65097326e-02,  1.11174431e-01,
         -3.06098230e-01,  9.83222440e-02,  2.98624126e-01,
          2.15278259e-01, -5.05930297e-01,  1.92694417e-02,
          4.14308498e-01, -4.36202990e-01, -4.74210380e-01,
          1.17493140e-02,  4.38124740e-01,  6.58042108e-01,
         -2.63660004e-01, -1.24679329e-01,  1.74983944e-01,
          2.39902358e-01, -1.48367980e-01, -5.88360133e-01,
          3.05081867e-01]]), array([[ 2.76389519e-01],
       [ 6.07843507e-01],
       [ 6.60426106e-01],
       [ 2.66574017e-01],
       [ 2.62473086e-01],
       [ 2.81850554e-01],
       [ 6.54327394e-02],
       [ 8.12970604e-01],
       [-1.25079269e-06],
       [ 4.56202634e-01],
       [ 2.76596595e-01],
       [-3.57482204e-01],
       [-1.62400963e-02],
       [ 5.57932748e-01],
       [-2.00486021e-03],
       [-1.83722180e-12],
       [ 4.42167169e-01],
       [-3.29044641e-01],
       [ 7.93532851e-01],
       [ 1.23137505e-01],
       [ 3.93343460e-01],
       [ 4.34479625e-01],
       [ 5.31863962e-01],
       [ 4.13317548e-01],
       [ 2.55747796e-01],
       [ 4.02248521e-01],
       [-5.68352120e-01],
       [ 5.79109331e-01],
       [ 4.01669458e-01],
       [ 4.93732390e-01],
       [-4.82040821e-16],
       [ 5.56265509e-01],
       [-5.80308756e-01],
       [-3.71609647e-01],
       [-5.08110212e-01],
       [-1.44175270e-01],
       [ 6.72855465e-01],
       [ 4.72891959e-01],
       [-6.11132704e-01],
       [ 5.69224432e-02],
       [-4.98312006e-01],
       [-2.93200346e-02],
       [-1.26471034e-01],
       [ 3.77016628e-01],
       [ 3.98564496e-01],
       [ 4.23424816e-01],
       [-2.15759805e-01],
       [-2.03893808e-03],
       [-4.45263988e-01],
       [-5.61266428e-01],
       [-2.98789688e-01],
       [-5.07166361e-01],
       [-8.42945108e-02],
       [-1.13045192e-01],
       [ 4.34398315e-01],
       [-6.84508456e-04],
       [ 7.51628166e-01],
       [ 5.05937849e-01],
       [-1.70958519e-02],
       [-4.64672912e-01],
       [-4.87293899e-01],
       [-2.29488757e-01],
       [-4.92455018e-01],
       [ 3.52605643e-01],
       [ 3.41836695e-01],
       [-4.49092793e-01],
       [-5.66171353e-02],
       [-4.50723508e-01],
       [ 7.07912849e-02],
       [ 3.60730272e-01],
       [-4.49775421e-01],
       [-4.07771998e-01],
       [ 6.38759475e-01],
       [-3.24467962e-04],
       [-4.05875879e-01],
       [ 5.97448930e-01],
       [ 3.50172572e-01],
       [ 5.98855834e-01],
       [-2.82812682e-01],
       [ 1.70577472e-01],
       [ 4.02802117e-01],
       [-7.06160384e-01],
       [ 4.05759751e-01],
       [ 7.07815310e-01],
       [ 5.59030802e-01],
       [-4.24147940e-01],
       [-3.26476559e-01],
       [ 2.50576707e-01],
       [-4.46338954e-01],
       [-7.33895485e-01],
       [-1.94376256e-01],
       [ 3.19262447e-01],
       [ 4.73907950e-01],
       [-2.56763088e-01],
       [-3.06131256e-01],
       [ 3.77109415e-01],
       [ 7.69463183e-01],
       [-5.45350463e-01],
       [-3.97744594e-01],
       [ 5.74659522e-01]])]
```

SVM

In [142…
```
svc = svm.SVC()
```

In [143…
```
svc_tennis = svc.fit(X_train_tennis,y_train_tennis)
```

In [144…
```
prediction_tennis_svm = svc_tennis.predict(X_test_tennis)
print("SVM: Tennis")
print("Accuracy:", metrics.accuracy_score(y_test_tennis, prediction_tennis_svm))
print("F1 Macro avg:", metrics.f1_score(y_test_tennis, prediction_tennis_svm, average="macro"))
print("F1 Weighted avg:", metrics.f1_score(y_test_tennis, prediction_tennis_svm, average="weighted"))
```

```
SVM: Tennis
Accuracy: 0.3333333333333333
F1 Macro avg: 0.25
F1 Weighted avg: 0.16666666666666666
```

In [145…
```
svc_cancer = svc.fit(X_train_cancer,y_train_cancer)
```

In [146…
```
prediction_cancer_svm = svc_cancer.predict(X_test_cancer)
print("SVM: Cancer")
print("Accuracy:", metrics.accuracy_score(y_test_cancer, prediction_cancer_svm))
print("F1 Macro avg:", metrics.f1_score(y_test_cancer, prediction_cancer_svm, average="macro"))
print("F1 Weighted avg:", metrics.f1_score(y_test_cancer, prediction_cancer_svm, average="macro"))
```

```
SVM: Cancer
Accuracy: 0.9385964912280702
F1 Macro avg: 0.9260220635950681
F1 Weighted avg: 0.9260220635950681
```

Analisis

Perbandingan Nilai Akurasi dan F1 Score dari masing-masing Algoritma pada dataset Play Tennis

Setelah melakukan pembelajaran dengan beberapa algoritma di atas, nilai tertinggi untuk Accuracy diperoleh dengan menggunakan algoritma K-Means dan LogisticRegression yaitu sebesar 0.66. Untuk untuk F1 Score, nilai tertinggi diperoleh dengan menggunakan algoritma Logistic Regression sebesar 0.66 untuk macro average F1 dan weighted average F1. Sedangkan nilai terendah untuk dataset Play Tennis didapat dengan keempat algoritma lainnya yaitu Decision Tree, Neural Network, ID3, dan SVM dengan skor 0.333 untuk Accuracy dan 0.25 / 0.167 untuk F1 Macro / Weighted average.

Perbandingan Nilai Akurasi dan F1 Score dari masing-masing Algoritma pada dataset Breast Cancer

Setelah melakukan pembelajaran dengan beberapa algoritma di atas, didapatkan nilai tertinggi dengan menggunakan Neural Network dan LogisticRegression dengan 0.956 untuk Accuracy, 0.948 untuk F1 Macro avg, dan 0.956 untuk F1 Weighted avg. Sedangkan yang terendah didapat dengan pembelajaran KMeans dengan 0.105 untuk Accuracy, 0.098 untuk F1 Macro avg, dan 0.066 untuk F1 Weighted avg.

Kesimpulan

Dengan melihat hasil atau nilai yang dihasilkan dari pembelajaran berbagai algoritma, kami melihat pola di mana skor yang dihasilkan untuk dataset breast cancer relatif lebih besar daripada untuk dataset play tennis. Hal ini kami asumsikan dikarenakan jumlah data yang sangat jauh berbeda di mana dataset breast cancer memiliki jumlah data yang banyak sedangkan data play tennis memiliki jumlah data yang sedikit sehingga proses pembelajaran untuk data play tennis kurang maksimal.