```
[24-1]  #task3
```

```
[24-2]  #import libraries
        import pandas as pd
        from sklearn.datasets import load_breast_cancer
        from sklearn.model_selection import train_test_split
        from sklearn.metrics import classification_report, accuracy_score,confusion_matrix
        import matplotlib.pyplot as plt
        from sklearn.tree import DecisionTreeClassifier, plot_tree
        from sklearn.ensemble import RandomForestClassifier
        import numpy as np
        from sklearn.preprocessing import StandardScaler

        # Load the breast cancer dataset
        cancer = load_breast_cancer()
```

```
[24-3]  df = pd.DataFrame(cancer.data, columns=cancer.feature_names)
        df['target'] = cancer.target

        # Display the first few rows of the dataframe
        print(df.head())
```

```
[24-4]  mean radius  mean texture  mean perimeter  mean area  mean smoothness  \
        0    17.99         10.38          122.80     1001.0          0.11840
        1    20.57         17.77          132.90     1326.0          0.08474
        2    19.69         21.25          130.00     1203.0          0.10960
        3    11.42         20.38          77.58      386.1          0.14250
        4    20.29         14.34          135.10     1297.0          0.10030

        mean compactness  mean concavity  mean concave points  mean symmetry  \
        0           0.27760          0.3001              0.14710         0.2419
        1           0.07864          0.0869              0.07017         0.1812
        2           0.15990          0.1974              0.12790         0.2069
        3           0.28390          0.2414              0.10520         0.2597
        4           0.13280          0.1980              0.10430         0.1809

        mean fractal dimension  ...  worst texture  worst perimeter  worst area  \
        0                 0.07871  ...          17.33           184.60      2019.0
        1                 0.05667  ...          23.41           158.80      1956.0
        2                 0.05999  ...          25.53           152.50      1709.0
        3                 0.09744  ...          26.50           98.87       567.7
        4                 0.05883  ...          16.67           152.20      1575.0

        worst smoothness  worst compactness  worst concavity  worst concave points  \
        0            0.1622             0.6656           0.7119                0.2654
        1            0.1238             0.1866           0.2416                0.1860
        2            0.1444             0.4245           0.4504                0.2430
        3            0.2098             0.8663           0.6869                0.2575
        4            0.1374             0.2050           0.4000                0.1625

        worst symmetry  worst fractal dimension  target
        0          0.4601                  0.1189       0
        1          0.2750                  0.08902      0
        2          0.3613                  0.08758      0
        3          0.6638                  0.173        0
        4          0.2364                  0.07678      0

        [5 rows x 31 columns]
```

```
[24-5]  print(df.info())
```

```
        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 569 entries, 0 to 568
        Data columns (total 31 columns):
         #   Column                   Non-Null Count  Dtype
        ---  ------                   --------------  -----
         0   mean radius              569 non-null    float64
         1   mean texture             569 non-null    float64
         2   mean perimeter           569 non-null    float64
         3   mean area                569 non-null    float64
         4   mean smoothness          569 non-null    float64
         5   mean compactness         569 non-null    float64
         6   mean concavity           569 non-null    float64
         7   mean concave points      569 non-null    float64
         8   mean symmetry            569 non-null    float64
         9   mean fractal dimension   569 non-null    float64
         10  radius error             569 non-null    float64
         11  texture error            569 non-null    float64
         12  perimeter error          569 non-null    float64
         13  area error               569 non-null    float64
         14  smoothness error         569 non-null    float64
         15  compactness error        569 non-null    float64
         16  concavity error          569 non-null    float64
         17  concave points error     569 non-null    float64
         18  symmetry error           569 non-null    float64
         19  fractal dimension error  569 non-null    float64
         20  worst radius             569 non-null    float64
         21  worst texture            569 non-null    float64
         22  worst perimeter          569 non-null    float64
         23  worst area               569 non-null    float64
         24  worst smoothness         569 non-null    float64
         25  worst compactness        569 non-null    float64
         26  worst concavity          569 non-null    float64
         27  worst concave points     569 non-null    float64
         28  worst symmetry           569 non-null    float64
         29  worst fractal dimension  569 non-null    float64
         30  target                   569 non-null    int32
        dtypes: float64(30), int32(1)
        memory usage: 138.7 KB
        None
```

```
[24-6]  print(df.describe())
```

```
                mean radius  mean texture  mean perimeter  mean area  \
        count    569.000000    569.000000      569.000000  569.000000
        mean      14.127292     19.289649       91.969033  654.889104
        std        3.524049      4.301036       24.298981  351.914129
        min        6.981000      9.710000       43.790000  143.500000
        25%       11.700000     16.170000       75.170000  420.300000
        50%       13.370000     18.840000       86.240000  551.100000
        75%       15.780000     21.080000      104.100000  782.700000
        max       28.110000     39.280000      188.500000 2501.000000

                mean smoothness  mean compactness  mean concavity  mean concave points  \
        count       569.000000        569.000000      569.000000           569.000000
        mean          0.096360          0.104341        0.088799             0.048919
        std           0.014064          0.052813        0.079720             0.038803
        min           0.052630          0.019380        0.000000             0.000000
        25%           0.086370          0.064920        0.029560             0.020310
        50%           0.095870          0.092630        0.061540             0.033500
        75%           0.105300          0.130400        0.130700             0.074000
        max           0.163400          0.345400        0.426800             0.201200

                mean symmetry  mean fractal dimension  ...  worst texture  \
        count     569.000000              569.000000  ...     569.000000
        mean       0.181162                0.062798  ...      25.677223
        std        0.027414                0.007060  ...       6.146258
        min        0.106000                0.049960  ...      12.020000
        25%        0.161900                0.057700  ...      21.080000
        50%        0.179200                0.061540  ...      25.410000
        75%        0.195700                0.066120  ...      29.720000
        max        0.304000                0.097440  ...      49.540000

                worst perimeter  worst area  worst smoothness  worst compactness  \
        count       569.000000  569.000000        569.000000         569.000000
        mean        107.261213  880.583128          0.132369           0.254265
        std          33.602542  569.356993          0.022832           0.157336
        min          50.410000  185.200000          0.071170           0.027290
        25%          84.110000  515.300000          0.116600           0.147200
        50%          97.660000  686.500000          0.131300           0.211900
        75%         125.400000  084.500000          0.146000           0.339100
        max         251.200000 4254.000000          0.222600           1.058000

                worst concavity  worst concave points  worst symmetry  \
        count       569.000000            569.000000      569.000000
        mean         0.272188              0.114606        0.290076
        std          0.208070              0.065732        0.061867
        min          0.000000              0.000000        0.156500
        25%          0.114500              0.064930        0.250400
        50%          0.226700              0.099930        0.282200
        75%          0.382900              0.161400        0.317500
        max          1.252000              0.291000        0.663800

                worst fractal dimension       target
        count               569.000000  569.000000
        mean                  0.083946    0.627417
        std                   0.018061    0.483918
        min                   0.055040    0.000000
        25%                   0.071460    0.000000
        50%                   0.080040    1.000000
        75%                   0.092080    1.000000
        max                   0.207500    1.000000

        [8 rows x 31 columns]
```

```
[24-7]  df.isnull().sum()
```

```
        mean radius                0
        mean texture               0
        mean perimeter             0
        mean area                  0
        mean smoothness            0
        mean compactness           0
        mean concavity             0
        mean concave points        0
        mean symmetry              0
        mean fractal dimension     0
        radius error               0
        texture error              0
        perimeter error            0
        area error                 0
        smoothness error           0
        compactness error          0
        concavity error            0
        concave points error       0
        symmetry error             0
        fractal dimension error    0
        worst radius               0
        worst texture              0
        worst perimeter            0
        worst area                 0
        worst smoothness           0
        worst compactness          0
        worst concavity            0
        worst concave points       0
        worst symmetry             0
        worst fractal dimension    0
        target                     0
        dtype: int64
```

```
[24-8]  # Split the data into features (X) and target (y)
        X = df.drop('target', axis=1)
        y = df['target']
```

```
[24-9]  # Split the data into training and testing sets
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
        X_train.shape,X_test.shape
```

```
[24-9]  ((455, 30), (114, 30))
```

```
[24-10] # Standardize the features
        scaler = StandardScaler()
        X_train_scaled = scaler.fit_transform(X_train)
        X_test_scaled = scaler.transform(X_test)
```

```
[24-11] # Convert the scaled arrays back to DataFrame
        X_train_scaled_df = pd.DataFrame(X_train_scaled, columns=X_train.columns)
        X_test_scaled_df = pd.DataFrame(X_test_scaled, columns=X_test.columns)

        # Now you can use the head() method
        print(X_train_scaled_df.head())
        print(X_test_scaled_df.head())
```

```
        mean radius  mean texture  mean perimeter  mean area  mean smoothness  \
        0    -1.440753     -0.035150       -1.362985  -1.136118         0.798573
        1     0.333333     -1.465216        0.330000   0.325000        -0.786667
        2     -1.350982     -1.240622       -1.345256  -1.109700        -1.525345
        3     -0.961797     1.416222       -0.962367  -0.860944         0.898598
        4     -1.117780     -0.010200       -1.120962  -0.945942         1.200511

        mean compactness  mean concavity  mean concave points  mean symmetry  \
        0          0.568           0.000             0.000000       0.000000
        1          2.405           2.411             2.411000       1.247600
        2         -0.907355      -0.903550            -0.906602       1.936855
        3          0.000000       0.000000             0.000000       0.000000
        4          -0.420802      -0.945941            -0.936600       3.264436

        mean fractal dimension  ...  worst radius  worst texture  worst perimeter  \
        0               0.000000  ...     -1.488475      -1.215961        -1.447507
        1               1.588200  ...      1.172534      1.151279         1.083837
        2              -0.907355  ...     -0.907355      -0.907355        -0.907355
        3               0.000000  ...      0.000000      0.000000         0.000000
        4               3.996513  ...     -1.085129      -1.334616        -1.117138

        worst area  worst smoothness  worst compactness  worst concavity  \
        0   -0.870364          0.000          0.000000         0.000000
        1    2.237405          0.793328       2.350001         1.939423
        2   -0.907355         -0.907355      -0.907355        -0.907355
        3    0.000000          0.000000       0.000000         0.000000
        4   -0.906549         -0.174876      -0.906879        -1.200146

        worst concave points  worst symmetry  worst fractal dimension  \
        0              0.000        0.000000                 0.000000
        1              2.237        1.936423                 1.936423
        2             -0.907       -0.907355                -0.907355
        3              0.000        0.000000                 0.000000
        4             -1.134        -0.174876                -1.200146

        [5 rows x 30 columns]
        mean radius  mean texture  mean perimeter  mean area  mean smoothness  \
        0    -0.404007     -1.228819       -0.384251  -0.385659         1.200656
        1     1.202302     -0.490604        1.206611   1.341471        -0.406229
        2     -0.333333     -1.465216       -0.330000  -0.325000         0.000000
        3     -0.400127     -0.351205       -0.410570  -0.399256         1.094246
        4     -0.729010     -1.113514       -0.706393  -0.705291         0.294612

        mean compactness  mean concavity  mean concave points  mean symmetry  \
        0          0.041606       -0.111405            -0.264869       0.431241
        1          1.202302        1.202302             1.202302       1.202302
        2          0.000414        0.729002             0.000722       0.423994
        3          0.533952       -0.144722            -0.523686       0.000646
        4          0.150599       -0.271202             0.107608       0.925934

        mean fractal dimension  ...  worst radius  worst texture  worst perimeter  \
        0               0.241262  ...     -0.404007      -1.228819        -0.384251
        1               2.588425  ...      1.202302      -0.490604         1.206611
        2               0.000722  ...     -0.333333      -1.465216        -0.330000
        3               0.646398  ...     -0.400127      -0.351205        -0.410570
        4               0.728524  ...     -0.284487      -0.452677

        worst area  worst smoothness  worst compactness  worst concavity  \
        0   -0.385659          1.200656          0.041606        -0.111405
        1    1.341471         -0.406229          0.533952        -0.144722
        2   -0.325000          0.000000          0.000414         0.729002
        3   -0.399256          1.094246          0.150599        -0.271202
        4   -0.728524          0.128718         -0.284487        -0.452677

        worst concave points  worst symmetry  worst fractal dimension
        0             -0.264869        0.431241                 0.241262
        1              1.202302        1.202302                 2.588425
        2              0.429277       -0.423394                 0.541322
        3              0.000000        0.000000                 0.000000
        4              0.952125        0.141780                 0.728524

        [5 rows x 30 columns]
```

```
[24-12] # Initialize and train a RandomForestClassifier
        rf = RandomForestClassifier(random_state=42)
        rf.fit(X_train_scaled, y_train)
```

```
[24-12] RandomForestClassifier
        RandomForestClassifier(random_state=42)
```

```
[24-13] # Make predictions on the test set
        y_pred = rf.predict(X_test_scaled)
        y_pred
```

```
[24-13] array([1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1])
```

```
[24-14] print("y_pred",y_pred)
        print("y_test",y_test)
```

```
        y_pred [1 0 0 1 1 0 0 0 0 1 1 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
        1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
        1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
        1 1 1 1]
        y_test 18    1
        499    0
        7      0
        235    1
        67     1
        548    1
        ..     ..
        465    1
        551    1
        100    0
        259    1
        294    1
        192    1
        239    1
        Name: target, Length: 114, dtype: int32
```

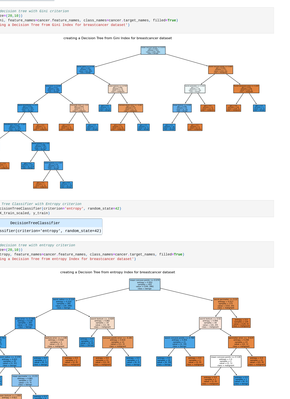
```
[24-15] # Evaluate the model
        print("Accuracy:", accuracy_score(y_test, y_pred))
        print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
        Accuracy: 0.9649122807017544
        Classification Report:
                      precision    recall  f1-score   support

                   0       0.98       0.93       0.95        42
                   1       0.96       0.99       0.97        72

            accuracy                            0.96       114
           macro avg       0.97       0.96       0.96       114
        weighted avg       0.97       0.96       0.96       114
```

```
[24-16] # Visualize one of the trees from the random forest
        plt.figure(figsize=(20,10))
        plot_tree(rf.estimators_[0], feature_names=cancer.feature_names, class_names=cancer.target_names, filled=True)
        plt.title("Decision Tree from Random Forest for Breast Cancer Dataset")
        plt.show()
```




```
[24-17] # Train Decision Tree Classifier with Gini criterion
        clf_gini = DecisionTreeClassifier(criterion='gini', random_state=42)
        clf_gini.fit(X_train_scaled, y_train)
```

```
[24-17] DecisionTreeClassifier
        DecisionTreeClassifier(random_state=42)
```

```
[24-18] # Visualize the decision tree with Gini criterion
        plt.figure(figsize=(20,10))
        plot_tree(clf_gini, feature_names=cancer.feature_names, class_names=cancer.target_names, filled=True)
        plt.title("Creating a Decision Tree from Gini Index for breastcancer dataset")
        plt.show()
```



```
[24-19] # Train Decision Tree Classifier with entropy criterion
        clf_entropy = DecisionTreeClassifier(criterion='entropy', random_state=42)
        clf_entropy.fit(X_train_scaled, y_train)
```

```
[24-19] DecisionTreeClassifier
        DecisionTreeClassifier(criterion='entropy', random_state=42)
```

```
[24-20] # Visualize the decision tree with entropy criterion
        plt.figure(figsize=(20,10))
        plot_tree(clf_entropy, feature_names=cancer.feature_names, class_names=cancer.target_names, filled=True)
        plt.title("creating a Decision Tree from entropy Index for breastcancer dataset")
        plt.show()
```



```
[24-21] # Make predictions
        y_pred_gini = clf_gini.predict(X_test_scaled)
        y_pred_entropy = clf_entropy.predict(X_test_scaled)
```

```
[24-22] # Evaluate the models
        print("Decision Tree with Gini Index")
        print("Accuracy:", accuracy_score(y_test, y_pred_gini))
        print("Classification Report:\n", classification_report(y_test, y_pred_gini))
        print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_gini))

        print("\nDecision Tree with Entropy")
        print("Accuracy:", accuracy_score(y_test, y_pred_entropy))
        print("Classification Report:\n", classification_report(y_test, y_pred_entropy))
        print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_entropy))
```

```
        Decision Tree with Gini Index
        Accuracy: 0.9473684210526315
        Classification Report:
                      precision    recall  f1-score   support

                   0       0.93       0.93       0.93        42
                   1       0.96       0.96       0.96        72

            accuracy                            0.95       114
           macro avg       0.94       0.94       0.94       114
        weighted avg       0.95       0.95       0.95       114

        Confusion Matrix:
         [[39  3]
         [ 3 69]]

        Decision Tree with Entropy
        Accuracy: 0.9649122807017544
        Classification Report:
                      precision    recall  f1-score   support

                   0       0.97       0.90       0.93        42
                   1       0.97       0.99       0.98        72

            accuracy                            0.96       114
           macro avg       0.96       0.94       0.95       114
        weighted avg       0.96       0.96       0.96       114

        Confusion Matrix:
         [[38  4]
         [ 1 71]]
```

```
[24-23] #RESULTS AND CONCLUSIONS
```

```
[24-24] # As this project, I built a DecisionTreeClassifier model to predict the breast cancer of Tumor.Using
        # the Breast Cancer Wisconsin dataset, the model achieved high accuracy on both the Gini and entropy criteria, with a 0.9474
        # In both the cases the training set accuracy was 1.0 indicating perfect accuracy score are almost same
        # The confusion matrix and classification report yields very good model performance
```