Assignment 3

Aim:- The aim of this assignment is to explore decision trees and their application in classification and regression.

Theory:- A decision tree is a supervised learning algorithm used for classification and regression. It works by recursively partitioning the data into smaller subsets based on the value of a feature. The algorithm selects the feature that provides the most information gain at each step. The result is a tree-like structure that can be used to make predictions.

The key concepts covered in this assignment would include:

* Understanding the difference between classification and regression trees.
* Learning how to measure the performance of a decision tree using metrics like accuracy and Mean Squared Error.
* Understanding how to handle overfitting and improve the performance of a decision tree using techniques like pruning.

Case study:- The dataset considered is named ‘diabetes\_dataset’. It has 768 rows and 9 columns. It contains columns having count of various measures like pregnancies (for ladies), glucose count of diabetic patients, their blood pressure, skin thickness, insulin measure, BMI, their age and whether they have diabetes or no.

*#Importing the required libraries*

**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

%matplotlib inline

data = pd.read\_csv("diabetes\_dataset.csv") print(data.shape)

print(data.head()) print(data.columns)

(768, 9)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Pregnancies | | Glucose | BloodPressure | SkinThickness | Insulin | |
| BMI \ | |  |  |  |  | |
| 0 | 6 | 148 | 72 | 35 | 0 | 33.6 |
| 1 | 1 | 85 | 66 | 29 | 0 | 26.6 |
| 2 | 8 | 183 | 64 | 0 | 0 | 23.3 |
| 3 | 1 | 89 | 66 | 23 | 94 | 28.1 |
| 4 | 0 | 137 | 40 | 35 | 168 | 43.1 |

DiabetesPedigreeFunction Age Outcome 0 0.627 50 1

1 0.351 31 0

2 0.672 32 1

3 0.167 21 0

4 2.288 33 1

Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',

'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'], dtype='object')

print(data.describe())

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pregnancies  Insulin \ | | Glucose | BloodPressure | SkinThickness |
| count 768.000000  768.000000 | | 768.000000 | 768.000000 | 768.000000 |
| mean | 3.845052 | 120.894531 | 69.105469 | 20.536458 |
| 79.799479  std | 3.369578 | 31.972618 | 19.355807 | 15.952218 |
| 115.244002  min | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0.000000 |  |  |  |  |
| 25% | 1.000000 | 99.000000 | 62.000000 | 0.000000 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0.000000 | |  | | | |
| 50% 3.000000 | | 117.000000 | 72.000000 | 23.000000 | |
| 30.500000 | |  |  |  | |
| 75% 6.000000 | | 140.250000 | 80.000000 | 32.000000 | |
| 127.250000  max 17.000000 | | 199.000000 | 122.000000 | 99.000000 | |
| 846.000000 | |  |  |  | |
|  | BMI | DiabetesPedigreeFunction | | Age | Outcome |
| count | 768.000000 | 768.000000 | | 768.000000 | 768.000000 |
| mean | 31.992578 | 0.471876 | | 33.240885 | 0.348958 |
| std | 7.884160 | 0.331329 | | 11.760232 | 0.476951 |
| min | 0.000000 | 0.078000 | | 21.000000 | 0.000000 |
| 25% | 27.300000 | 0.243750 | | 24.000000 | 0.000000 |
| 50% | 32.000000 | 0.372500 | | 29.000000 | 0.000000 |
| 75% | 36.600000 | 0.626250 | | 41.000000 | 1.000000 |
| max | 67.100000 | 2.420000 | | 81.000000 | 1.000000 |

data.isnull().sum()

Pregnancies 0

Glucose 0

BloodPressure 0

SkinThickness 0

Insulin 0

BMI 0

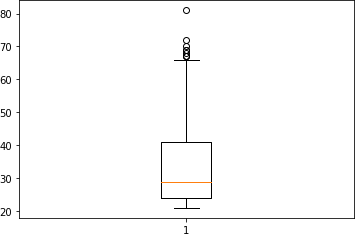
DiabetesPedigreeFunction 0

Age 0

Outcome 0

dtype: int64

*#Checking outlires through boxplot* plt.boxplot(data['Age']) plt.show()



*#Checking Outlier by definition and treating outliers*

*#getting median Age*

Age\_col\_df = pd.DataFrame(data['Age']) Age\_median = Age\_col\_df.median()

*#getting IQR of Age column*

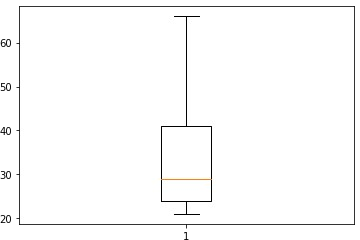
Q3 = Age\_col\_df.quantile(q=0.75) Q1 = Age\_col\_df.quantile(q=0.25)

IQR = Q3-Q1

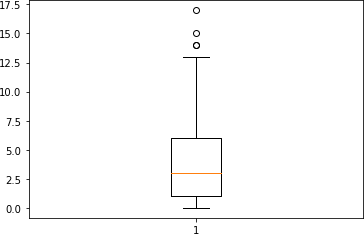
*#Deriving boundaries of Outliers* IQR\_LL = int(Q1 - 1.5\*IQR) IQR\_UL = int(Q3 + 1.5\*IQR)

*#Finding and treating outliers - both lower and upper end* data.loc[data['Age']>IQR\_UL , 'Age'] = int(Age\_col\_df.quantile(q=0.90)) data.loc[data['Age']<IQR\_LL , 'Age'] = int(Age\_col\_df.quantile(q=0.01))

*#Check max age value now* max(data['Age']) plt.boxplot(data['Age']) plt.show()



*#Checking outlires through boxplot* plt.boxplot(data['Pregnancies']) plt.show()



*#Checking Outlier by definition and treating outliers #getting median Age*

Age\_col\_df = pd.DataFrame(data['Pregnancies']) Age\_median = Age\_col\_df.median()

*#getting IQR of Age column*

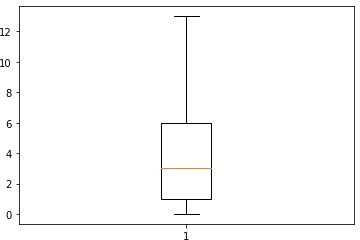
Q3 = Age\_col\_df.quantile(q=0.75) Q1 = Age\_col\_df.quantile(q=0.25)

IQR = Q3-Q1

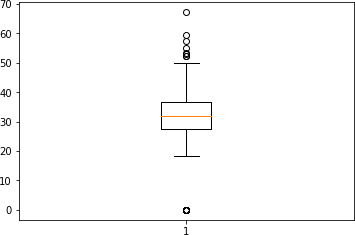
*#Deriving boundaries of Outliers* IQR\_LL = int(Q1 - 1.5\*IQR) IQR\_UL = int(Q3 + 1.5\*IQR)

*#Finding and treating outliers - both lower and upper end* data.loc[data['Pregnancies']>IQR\_UL , 'Pregnancies'] = int(Age\_col\_df.quantile(q=0.90)) data.loc[data['Pregnancies']<IQR\_LL , 'Pregnancies'] = int(Age\_col\_df.quantile(q=0.01))

*#Check max value now* max(data['Pregnancies']) plt.boxplot(data['Pregnancies']) plt.show()



*#Checking outlires through boxplot* plt.boxplot(data['BMI']) plt.show()



*#Checking Outlier by definition and treating outliers*

*#getting median Age*

Age\_col\_df = pd.DataFrame(data['BMI']) Age\_median = Age\_col\_df.median()

*#getting IQR of Age column*

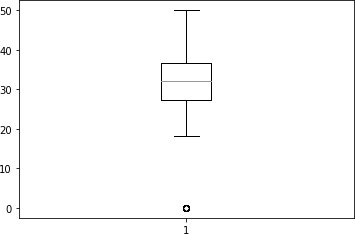
Q3 = Age\_col\_df.quantile(q=0.75) Q1 = Age\_col\_df.quantile(q=0.25)

IQR = Q3-Q1

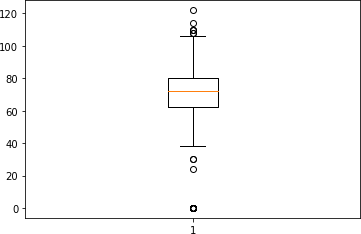
*#Deriving boundaries of Outliers* IQR\_LL = int(Q1 - 1.5\*IQR) IQR\_UL = int(Q3 + 1.5\*IQR)

*#Finding and treating outliers - both lower and upper end* data.loc[data['BMI']>IQR\_UL , 'BMI'] = int(Age\_col\_df.quantile(q=0.90)) data.loc[data['BMI']<IQR\_LL , 'BMI'] = int(Age\_col\_df.quantile(q=0.01))

*#Check max value now* max(data['BMI']) plt.boxplot(data['BMI']) plt.show()



*#Checking outlires through boxplot* plt.boxplot(data['BloodPressure']) plt.show()



*#Checking Outlier by definition and treating outliers #getting median Age*

Age\_col\_df = pd.DataFrame(data['BloodPressure']) Age\_median = Age\_col\_df.median()

*#getting IQR of Age column*

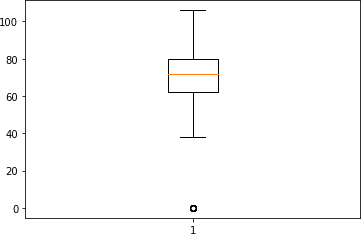
Q3 = Age\_col\_df.quantile(q=0.75) Q1 = Age\_col\_df.quantile(q=0.25)

IQR = Q3-Q1

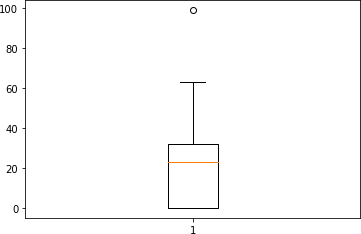
*#Deriving boundaries of Outliers* IQR\_LL = int(Q1 - 1.5\*IQR) IQR\_UL = int(Q3 + 1.5\*IQR)

*#Finding and treating outliers - both lower and upper end* data.loc[data['BloodPressure']>IQR\_UL , 'BloodPressure'] = int(Age\_col\_df.quantile(q=0.90)) data.loc[data['BloodPressure']<IQR\_LL , 'BloodPressure'] = int(Age\_col\_df.quantile(q=0.01))

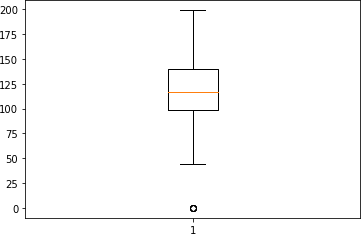
*#Check max value now* max(data['BloodPressure']) plt.boxplot(data['BloodPressure']) plt.show()



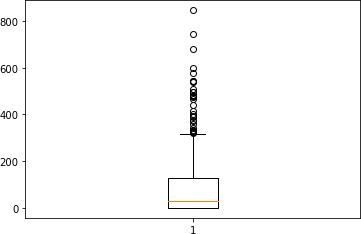
*#Checking outlires through boxplot* plt.boxplot(data['SkinThickness']) plt.show()



*#Checking outlires through boxplot* plt.boxplot(data['Glucose']) plt.show()



*#Checking outlires through boxplot* plt.boxplot(data['Insulin']) plt.show()



*#Checking Outlier by definition and treating outliers*

*#getting median Age*

Age\_col\_df = pd.DataFrame(data['Insulin']) Age\_median = Age\_col\_df.median()

*#getting IQR of Age column*

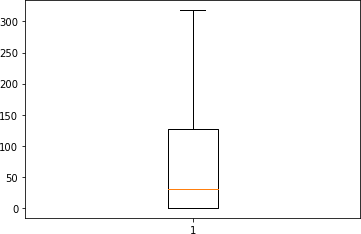
Q3 = Age\_col\_df.quantile(q=0.75) Q1 = Age\_col\_df.quantile(q=0.25)

IQR = Q3-Q1

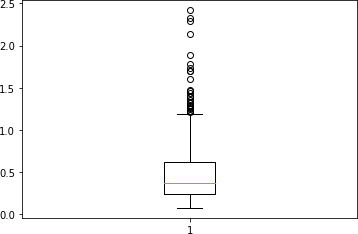
*#Deriving boundaries of Outliers* IQR\_LL = int(Q1 - 1.5\*IQR) IQR\_UL = int(Q3 + 1.5\*IQR)

*#Finding and treating outliers - both lower and upper end* data.loc[data['Insulin']>IQR\_UL , 'Insulin'] = int(Age\_col\_df.quantile(q=0.90)) data.loc[data['Insulin']<IQR\_LL , 'Insulin'] = int(Age\_col\_df.quantile(q=0.01))

*#Check max value now* max(data['Insulin']) plt.boxplot(data['Insulin']) plt.show()



*#Checking outlires through boxplot* plt.boxplot(data['DiabetesPedigreeFunction']) plt.show()



*#Checking Outlier by definition and treating outliers #getting median Age*

Age\_col\_df = pd.DataFrame(data['DiabetesPedigreeFunction']) Age\_median = Age\_col\_df.median()

*#getting IQR of Age column*

Q3 = Age\_col\_df.quantile(q=0.75) Q1 = Age\_col\_df.quantile(q=0.25)

IQR = Q3-Q1

*#Deriving boundaries of Outliers* IQR\_LL = int(Q1 - 1.5\*IQR) IQR\_UL = int(Q3 + 1.5\*IQR)

*#Finding and treating outliers - both lower and upper end* data.loc[data['DiabetesPedigreeFunction']>IQR\_UL , 'DiabetesPedigreeFunction'] = int(Age\_col\_df.quantile(q=0.90)) data.loc[data['DiabetesPedigreeFunction']<IQR\_LL , 'DiabetesPedigr'] = int(Age\_col\_df.quantile(q=0.01))

feature\_cols = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'Age']

X = data[feature\_cols]

data['Outcome'] = np.where(data['Outcome']>=0.9,1,0) Y = data['Outcome']

**from** sklearn.tree **import** DecisionTreeClassifier

**from** sklearn.model\_selection **import** cross\_val\_score **from** sklearn.model\_selection **import** train\_test\_split **from** sklearn **import** metrics

*#Splitting the dataset into training set and testing set*

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size = 0.3)

*#Implementing the decision tree classifier*

clf = DecisionTreeClassifier()

clf = clf.fit(x\_train, y\_train) *#Training the model* y\_pred = clf.predict(x\_test) *#Testing the model* Y\_train\_pred = clf.predict(x\_train)

*#Checking the training and testing accuracy*

print('Training Accuracy = ', metrics.accuracy\_score(y\_train,

Y\_train\_pred)) *#Training Accuracy*

print('Testing Accuracy = ', metrics.accuracy\_score(y\_test, y\_pred))

*#Testing Accuracy*

*#Checking the cross validation score*

clf1 = DecisionTreeClassifier()

print('Average Cross Validation Score = ', cross\_val\_score(clf1, x\_train, y\_train, cv = 10, scoring = 'accuracy').mean())

Training Accuracy = 1.0

Testing Accuracy = 0.7142857142857143

Average Cross Validation Score = 0.7077917540181692

**from** sklearn **import** tree tree.plot\_tree(clf)

[Text(0.6091958142201835, 0.96875, 'X[1] <= 157.5\ngini = 0.448\

nsamples = 537\nvalue = [355, 182]'),

Text(0.25967603211009177, 0.90625, 'X[5] <= 27.85\ngini = 0.38\

nsamples = 459\nvalue = [342, 117]'),

Text(0.07110091743119266, 0.84375, 'X[1] <= 105.5\ngini = 0.131\

nsamples = 142\nvalue = [132, 10]'),

Text(0.06192660550458716, 0.78125, 'gini = 0.0\nsamples = 65\nvalue = [65, 0]'),

Text(0.08027522935779817, 0.78125, 'X[5] <= 27.05\ngini = 0.226\

nsamples = 77\nvalue = [67, 10]'),

Text(0.04128440366972477, 0.71875, 'X[5] <= 9.8\ngini = 0.168\

nsamples = 65\nvalue = [59, 6]'),

Text(0.01834862385321101, 0.65625, 'X[0] <= 7.0\ngini = 0.5\nsamples

= 4\nvalue = [2, 2]'),

Text(0.009174311926605505, 0.59375, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),

Text(0.027522935779816515, 0.59375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),

Text(0.06422018348623854, 0.65625, 'X[1] <= 107.5\ngini = 0.123\

nsamples = 61\nvalue = [57, 4]'),

Text(0.045871559633027525, 0.59375, 'X[2] <= 65.0\ngini = 0.408\

nsamples = 7\nvalue = [5, 2]'),

Text(0.03669724770642202, 0.53125, 'X[0] <= 1.5\ngini = 0.444\

nsamples = 3\nvalue = [1, 2]'),

Text(0.027522935779816515, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.045871559633027525, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),

Text(0.05504587155963303, 0.53125, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),

Text(0.08256880733944955, 0.59375, 'X[1] <= 133.5\ngini = 0.071\

nsamples = 54\nvalue = [52, 2]'),

Text(0.07339449541284404, 0.53125, 'gini = 0.0\nsamples = 40\nvalue = [40, 0]'),

Text(0.09174311926605505, 0.53125, 'X[3] <= 8.5\ngini = 0.245\

nsamples = 14\nvalue = [12, 2]'),

Text(0.08256880733944955, 0.46875, 'X[1] <= 134.5\ngini = 0.444\

nsamples = 6\nvalue = [4, 2]'),

Text(0.07339449541284404, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),

Text(0.09174311926605505, 0.40625, 'X[6] <= 25.5\ngini = 0.32\

nsamples = 5\nvalue = [4, 1]'),

Text(0.08256880733944955, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),

Text(0.10091743119266056, 0.34375, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),

Text(0.10091743119266056, 0.46875, 'gini = 0.0\nsamples = 8\nvalue =

[8, 0]'),

Text(0.11926605504587157, 0.71875, 'X[2] <= 72.0\ngini = 0.444\

nsamples = 12\nvalue = [8, 4]'),

Text(0.11009174311926606, 0.65625, 'X[3] <= 25.0\ngini = 0.32\

nsamples = 5\nvalue = [1, 4]'),

Text(0.10091743119266056, 0.59375, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),

Text(0.11926605504587157, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.12844036697247707, 0.65625, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),

Text(0.4482511467889908, 0.84375, 'X[6] <= 30.5\ngini = 0.447\

nsamples = 317\nvalue = [210, 107]'),

Text(0.2408256880733945, 0.78125, 'X[2] <= 22.0\ngini = 0.336\

nsamples = 173\nvalue = [136, 37]'),

Text(0.1559633027522936, 0.71875, 'X[1] <= 117.0\ngini = 0.408\

nsamples = 7\nvalue = [2, 5]'),

Text(0.14678899082568808, 0.65625, 'X[5] <= 38.15\ngini = 0.444\

nsamples = 3\nvalue = [2, 1]'),

Text(0.13761467889908258, 0.59375, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),

Text(0.1559633027522936, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),

Text(0.1651376146788991, 0.65625, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),

Text(0.3256880733944954, 0.71875, 'X[1] <= 119.5\ngini = 0.311\

nsamples = 166\nvalue = [134, 32]'),

Text(0.22247706422018348, 0.65625, 'X[3] <= 30.5\ngini = 0.227\

nsamples = 107\nvalue = [93, 14]'),

Text(0.1743119266055046, 0.59375, 'X[5] <= 36.35\ngini = 0.114\

nsamples = 66\nvalue = [62, 4]'),

Text(0.1559633027522936, 0.53125, 'X[6] <= 29.5\ngini = 0.038\

nsamples = 52\nvalue = [51, 1]'),

Text(0.14678899082568808, 0.46875, 'gini = 0.0\nsamples = 49\nvalue = [49, 0]'),

Text(0.1651376146788991, 0.46875, 'X[0] <= 4.5\ngini = 0.444\nsamples

= 3\nvalue = [2, 1]'),

Text(0.1559633027522936, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),

Text(0.1743119266055046, 0.40625, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),

Text(0.1926605504587156, 0.53125, 'X[5] <= 36.7\ngini = 0.337\

nsamples = 14\nvalue = [11, 3]'),

Text(0.1834862385321101, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),

Text(0.2018348623853211, 0.46875, 'X[2] <= 79.0\ngini = 0.26\nsamples

= 13\nvalue = [11, 2]'),

Text(0.1926605504587156, 0.40625, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),

Text(0.21100917431192662, 0.40625, 'X[1] <= 93.5\ngini = 0.48\

nsamples = 5\nvalue = [3, 2]'),

Text(0.2018348623853211, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),

Text(0.22018348623853212, 0.34375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),

Text(0.2706422018348624, 0.59375, 'X[0] <= 1.5\ngini = 0.369\nsamples

= 41\nvalue = [31, 10]'),

Text(0.24770642201834864, 0.53125, 'X[5] <= 34.05\ngini = 0.165\

nsamples = 22\nvalue = [20, 2]'),

Text(0.23853211009174313, 0.46875, 'X[5] <= 33.45\ngini = 0.444\

nsamples = 6\nvalue = [4, 2]'),

Text(0.22935779816513763, 0.40625, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),

Text(0.24770642201834864, 0.40625, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),

Text(0.25688073394495414, 0.46875, 'gini = 0.0\nsamples = 16\nvalue = [16, 0]'),

Text(0.29357798165137616, 0.53125, 'X[4] <= 62.0\ngini = 0.488\

nsamples = 19\nvalue = [11, 8]'),

Text(0.27522935779816515, 0.46875, 'X[5] <= 37.95\ngini = 0.219\

nsamples = 8\nvalue = [7, 1]'),

Text(0.26605504587155965, 0.40625, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),

Text(0.28440366972477066, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),

Text(0.3119266055045872, 0.46875, 'X[1] <= 110.5\ngini = 0.463\

nsamples = 11\nvalue = [4, 7]'),

Text(0.30275229357798167, 0.40625, 'X[2] <= 60.0\ngini = 0.219\

nsamples = 8\nvalue = [1, 7]'),

Text(0.29357798165137616, 0.34375, 'X[0] <= 2.5\ngini = 0.5\nsamples

= 2\nvalue = [1, 1]'),

Text(0.28440366972477066, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.30275229357798167, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),

Text(0.3119266055045872, 0.34375, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),

Text(0.3211009174311927, 0.40625, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),

Text(0.4288990825688073, 0.65625, 'X[2] <= 72.0\ngini = 0.424\

nsamples = 59\nvalue = [41, 18]'),

Text(0.38073394495412843, 0.59375, 'X[0] <= 0.5\ngini = 0.5\nsamples

= 26\nvalue = [13, 13]'),

Text(0.37155963302752293, 0.53125, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),

Text(0.38990825688073394, 0.53125, 'X[6] <= 24.5\ngini = 0.491\

nsamples = 23\nvalue = [13, 10]'),

Text(0.3577981651376147, 0.46875, 'X[2] <= 68.0\ngini = 0.375\

nsamples = 12\nvalue = [9, 3]'),

Text(0.3394495412844037, 0.40625, 'X[5] <= 40.25\ngini = 0.198\

nsamples = 9\nvalue = [8, 1]'),

Text(0.3302752293577982, 0.34375, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),

Text(0.3486238532110092, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),

Text(0.3761467889908257, 0.40625, 'X[4] <= 47.5\ngini = 0.444\

nsamples = 3\nvalue = [1, 2]'),

Text(0.3669724770642202, 0.34375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),

Text(0.3853211009174312, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.42201834862385323, 0.46875, 'X[5] <= 39.05\ngini = 0.463\

nsamples = 11\nvalue = [4, 7]'),

Text(0.41284403669724773, 0.40625, 'X[0] <= 3.5\ngini = 0.346\

nsamples = 9\nvalue = [2, 7]'),

Text(0.4036697247706422, 0.34375, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),

Text(0.42201834862385323, 0.34375, 'X[2] <= 69.0\ngini = 0.444\

nsamples = 3\nvalue = [2, 1]'),

Text(0.41284403669724773, 0.28125, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),

Text(0.43119266055045874, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),

Text(0.43119266055045874, 0.40625, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),

Text(0.47706422018348627, 0.59375, 'X[2] <= 85.5\ngini = 0.257\

nsamples = 33\nvalue = [28, 5]'),

Text(0.44954128440366975, 0.53125, 'X[0] <= 3.5\ngini = 0.08\nsamples

= 24\nvalue = [23, 1]'),

Text(0.44036697247706424, 0.46875, 'gini = 0.0\nsamples = 19\nvalue = [19, 0]'),

Text(0.45871559633027525, 0.46875, 'X[3] <= 12.5\ngini = 0.32\

nsamples = 5\nvalue = [4, 1]'),

Text(0.44954128440366975, 0.40625, 'X[1] <= 131.0\ngini = 0.5\

nsamples = 2\nvalue = [1, 1]'),

Text(0.44036697247706424, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),

Text(0.45871559633027525, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.46788990825688076, 0.40625, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),

Text(0.5045871559633027, 0.53125, 'X[4] <= 137.5\ngini = 0.494\

nsamples = 9\nvalue = [5, 4]'),

Text(0.4954128440366973, 0.46875, 'X[6] <= 28.0\ngini = 0.32\nsamples

= 5\nvalue = [1, 4]'),

Text(0.48623853211009177, 0.40625, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),

Text(0.5045871559633027, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.5137614678899083, 0.46875, 'gini = 0.0\nsamples = 4\nvalue =

[4, 0]'),

Text(0.6556766055045872, 0.78125, 'X[1] <= 94.5\ngini = 0.5\nsamples

= 144\nvalue = [74, 70]'),

Text(0.5504587155963303, 0.71875, 'X[6] <= 53.5\ngini = 0.269\

nsamples = 25\nvalue = [21, 4]'),

Text(0.5412844036697247, 0.65625, 'X[1] <= 28.5\ngini = 0.219\

nsamples = 24\nvalue = [21, 3]'),

Text(0.5321100917431193, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),

Text(0.5504587155963303, 0.59375, 'X[5] <= 29.75\ngini = 0.159\

nsamples = 23\nvalue = [21, 2]'),

Text(0.5412844036697247, 0.53125, 'X[4] <= 35.5\ngini = 0.5\nsamples

= 4\nvalue = [2, 2]'),

Text(0.5321100917431193, 0.46875, 'X[6] <= 31.5\ngini = 0.444\

nsamples = 3\nvalue = [1, 2]'),

Text(0.5229357798165137, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.5412844036697247, 0.40625, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),

Text(0.5504587155963303, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.5596330275229358, 0.53125, 'gini = 0.0\nsamples = 19\nvalue = [19, 0]'),

Text(0.5596330275229358, 0.65625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),

Text(0.760894495412844, 0.71875, 'X[5] <= 40.75\ngini = 0.494\

nsamples = 119\nvalue = [53, 66]'),

Text(0.6685779816513762, 0.65625, 'X[0] <= 1.5\ngini = 0.5\nsamples =

102\nvalue = [51, 51]'),

Text(0.5871559633027523, 0.59375, 'X[2] <= 30.0\ngini = 0.32\nsamples

= 15\nvalue = [3, 12]'),

Text(0.5779816513761468, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.5963302752293578, 0.53125, 'X[4] <= 190.0\ngini = 0.245\

nsamples = 14\nvalue = [2, 12]'),

Text(0.5871559633027523, 0.46875, 'X[2] <= 91.0\ngini = 0.142\

nsamples = 13\nvalue = [1, 12]'),

Text(0.5779816513761468, 0.40625, 'gini = 0.0\nsamples = 10\nvalue = [0, 10]'),

Text(0.5963302752293578, 0.40625, 'X[3] <= 14.0\ngini = 0.444\

nsamples = 3\nvalue = [1, 2]'),

Text(0.5871559633027523, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.6055045871559633, 0.34375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),

Text(0.6055045871559633, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.75, 0.59375, 'X[2] <= 83.0\ngini = 0.495\nsamples = 87\nvalue

= [48, 39]'),

Text(0.6926605504587156, 0.53125, 'X[2] <= 75.5\ngini = 0.499\

nsamples = 65\nvalue = [31, 34]'),

Text(0.6513761467889908, 0.46875, 'X[3] <= 19.0\ngini = 0.478\

nsamples = 38\nvalue = [23, 15]'),

Text(0.6330275229357798, 0.40625, 'X[6] <= 47.0\ngini = 0.494\

nsamples = 18\nvalue = [8, 10]'),

Text(0.6238532110091743, 0.34375, 'X[0] <= 3.5\ngini = 0.444\nsamples

= 15\nvalue = [5, 10]'),

Text(0.6146788990825688, 0.28125, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),

Text(0.6330275229357798, 0.28125, 'X[0] <= 9.5\ngini = 0.355\nsamples

= 13\nvalue = [3, 10]'),

Text(0.6238532110091743, 0.21875, 'X[1] <= 100.0\ngini = 0.278\

nsamples = 12\nvalue = [2, 10]'),

Text(0.6055045871559633, 0.15625, 'X[2] <= 69.0\ngini = 0.5\nsamples

= 2\nvalue = [1, 1]'),

Text(0.5963302752293578, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),

Text(0.6146788990825688, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.6422018348623854, 0.15625, 'X[6] <= 34.5\ngini = 0.18\nsamples

= 10\nvalue = [1, 9]'),

Text(0.6330275229357798, 0.09375, 'X[6] <= 33.5\ngini = 0.444\

nsamples = 3\nvalue = [1, 2]'),

Text(0.6238532110091743, 0.03125, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),

Text(0.6422018348623854, 0.03125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.6513761467889908, 0.09375, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]'),

Text(0.6422018348623854, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.6422018348623854, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),

Text(0.6697247706422018, 0.40625, 'X[5] <= 38.55\ngini = 0.375\

nsamples = 20\nvalue = [15, 5]'),

Text(0.6605504587155964, 0.34375, 'X[6] <= 39.5\ngini = 0.278\

nsamples = 18\nvalue = [15, 3]'),

Text(0.6513761467889908, 0.28125, 'gini = 0.0\nsamples = 10\nvalue = [10, 0]'),

|  |  |  |  |
| --- | --- | --- | --- |
| Text(0.6697247706422018,  nsamples = 8\nvalue = [5, Text(0.6605504587155964, | 0.28125,  3]'),  0.21875, | 'X[5]  'gini | <= 33.85\ngini = 0.469\  = 0.0\nsamples = 3\nvalue = |
| [0, 3]'), |  |  |  |
| Text(0.6788990825688074, | 0.21875, | 'gini | = 0.0\nsamples = 5\nvalue = |
| [5, 0]'), |  |  |  |
| Text(0.6788990825688074, | 0.34375, | 'gini | = 0.0\nsamples = 2\nvalue = |
| [0, 2]'), |  |  |  |
| Text(0.7339449541284404, | 0.46875, | 'X[0] | <= 5.5\ngini = 0.417\nsamples |
| = 27\nvalue = [8, 19]'), |  |  |  |
| Text(0.7064220183486238, | 0.40625, | 'X[2] | <= 77.0\ngini = 0.496\ |

|  |  |  |  |
| --- | --- | --- | --- |
| nsamples = 11\nvalue = [6, 5]'), | | | |
| Text(0.6972477064220184, | 0.34375, | 'gini | = 0.0\nsamples = 2\nvalue = |
| [0, 2]'), |  |  |  |
| Text(0.7155963302752294, | 0.34375, | 'X[1] | <= 128.0\ngini = 0.444\ |
| nsamples = 9\nvalue = [6, | 3]'), |  |  |
| Text(0.7064220183486238, | 0.28125, | 'gini | = 0.0\nsamples = 4\nvalue = |
| [4, 0]'), |  |  |  |
| Text(0.7247706422018348, | 0.28125, | 'X[6] | <= 60.5\ngini = 0.48\nsamples |
| = 5\nvalue = [2, 3]'), |  |  |  |
| Text(0.7155963302752294, | 0.21875, | 'gini | = 0.0\nsamples = 3\nvalue = |
| [0, 3]'), |  |  |  |
| Text(0.7339449541284404, | 0.21875, | 'gini | = 0.0\nsamples = 2\nvalue = |
| [2, 0]'), |  |  |  |
| Text(0.7614678899082569, 0.40625,  nsamples = 16\nvalue = [2, 14]'), | | 'X[4] | <= 107.0\ngini = 0.219\ |
| Text(0.7522935779816514, | 0.34375, | 'X[0] | <= 7.5\ngini = 0.408\nsamples |
| = 7\nvalue = [2, 5]'), |  |  |  |
| Text(0.7431192660550459, | 0.28125, | 'gini | = 0.0\nsamples = 4\nvalue = |
| [0, 4]'), |  |  |  |
| Text(0.7614678899082569, | 0.28125, | 'X[3] | <= 33.5\ngini = 0.444\ |
| nsamples = 3\nvalue = [2, | 1]'), |  |  |
| Text(0.7522935779816514, | 0.21875, | 'gini | = 0.0\nsamples = 2\nvalue = |
| [2, 0]'), |  |  |  |
| Text(0.7706422018348624, | 0.21875, | 'gini | = 0.0\nsamples = 1\nvalue = |
| [0, 1]'), |  |  |  |
| Text(0.7706422018348624, | 0.34375, | 'gini | = 0.0\nsamples = 9\nvalue = |
| [0, 9]'), |  |  |  |

Text(0.8073394495412844, 0.53125, 'X[4] <= 143.0\ngini = 0.351\

nsamples = 22\nvalue = [17, 5]'),

Text(0.7889908256880734, 0.46875, 'X[3] <= 39.5\ngini = 0.111\

nsamples = 17\nvalue = [16, 1]'),

Text(0.7798165137614679, 0.40625, 'gini = 0.0\nsamples = 16\nvalue = [16, 0]'),

Text(0.7981651376146789, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),

Text(0.8256880733944955, 0.46875, 'X[1] <= 151.5\ngini = 0.32\

nsamples = 5\nvalue = [1, 4]'),

Text(0.8165137614678899, 0.40625, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),

Text(0.8348623853211009, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.8532110091743119, 0.65625, 'X[3] <= 53.5\ngini = 0.208\

nsamples = 17\nvalue = [2, 15]'),

Text(0.8440366972477065, 0.59375, 'X[6] <= 46.5\ngini = 0.117\

nsamples = 16\nvalue = [1, 15]'),

Text(0.8348623853211009, 0.53125, 'gini = 0.0\nsamples = 14\nvalue = [0, 14]'),

Text(0.8532110091743119, 0.53125, 'X[6] <= 47.5\ngini = 0.5\nsamples

= 2\nvalue = [1, 1]'),

Text(0.8440366972477065, 0.46875, 'gini = 0.0\nsamples = 1\nvalue =

[1, 0]'),

Text(0.8623853211009175, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),

Text(0.8623853211009175, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.9587155963302753, 0.90625, 'X[5] <= 46.1\ngini = 0.278\

nsamples = 78\nvalue = [13, 65]'),

Text(0.9357798165137615, 0.84375, 'X[2] <= 92.0\ngini = 0.25\nsamples

= 75\nvalue = [11, 64]'),

Text(0.9174311926605505, 0.78125, 'X[5] <= 36.75\ngini = 0.219\

nsamples = 72\nvalue = [9, 63]'),

Text(0.908256880733945, 0.71875, 'X[5] <= 36.6\ngini = 0.315\nsamples

= 46\nvalue = [9, 37]'),

Text(0.8990825688073395, 0.65625, 'X[3] <= 26.5\ngini = 0.292\

nsamples = 45\nvalue = [8, 37]'),

Text(0.8807339449541285, 0.59375, 'X[6] <= 27.0\ngini = 0.384\

nsamples = 27\nvalue = [7, 20]'),

Text(0.8715596330275229, 0.53125, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),

Text(0.8899082568807339, 0.53125, 'X[6] <= 33.5\ngini = 0.32\nsamples

= 25\nvalue = [5, 20]'),

Text(0.8807339449541285, 0.46875, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),

Text(0.8990825688073395, 0.46875, 'X[6] <= 34.5\ngini = 0.415\

nsamples = 17\nvalue = [5, 12]'),

Text(0.8899082568807339, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.908256880733945, 0.40625, 'X[0] <= 5.5\ngini = 0.375\nsamples

= 16\nvalue = [4, 12]'),

Text(0.8990825688073395, 0.34375, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),

Text(0.9174311926605505, 0.34375, 'X[2] <= 82.0\ngini = 0.463\

nsamples = 11\nvalue = [4, 7]'),

Text(0.908256880733945, 0.28125, 'X[5] <= 25.8\ngini = 0.5\nsamples =

8\nvalue = [4, 4]'),

Text(0.8990825688073395, 0.21875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),

Text(0.9174311926605505, 0.21875, 'X[1] <= 164.5\ngini = 0.444\

nsamples = 6\nvalue = [4, 2]'),

Text(0.908256880733945, 0.15625, 'X[2] <= 65.0\ngini = 0.444\nsamples

= 3\nvalue = [1, 2]'),

Text(0.8990825688073395, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.9174311926605505, 0.09375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),

Text(0.926605504587156, 0.15625, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),

Text(0.926605504587156, 0.28125, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),

Text(0.9174311926605505, 0.59375, 'X[5] <= 29.5\ngini = 0.105\

nsamples = 18\nvalue = [1, 17]'),

Text(0.908256880733945, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.926605504587156, 0.53125, 'gini = 0.0\nsamples = 17\nvalue = [0, 17]'),

Text(0.9174311926605505, 0.65625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),

Text(0.926605504587156, 0.71875, 'gini = 0.0\nsamples = 26\nvalue = [0, 26]'),

Text(0.9541284403669725, 0.78125, 'X[2] <= 99.5\ngini = 0.444\

nsamples = 3\nvalue = [2, 1]'),

Text(0.944954128440367, 0.71875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),

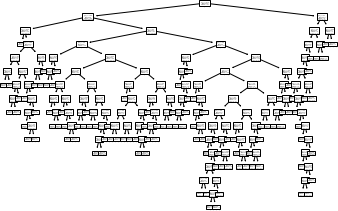
Text(0.963302752293578, 0.71875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),

Text(0.981651376146789, 0.84375, 'X[1] <= 180.5\ngini = 0.444\

nsamples = 3\nvalue = [2, 1]'),

Text(0.9724770642201835, 0.78125, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),

Text(0.9908256880733946, 0.78125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]')]



**import** graphviz

**from** sklearn.tree **import** export\_graphviz

**from** six **import** StringIO

**from** IPython.display **import** Image

**import** pydotplus

dot\_data =StringIO()

*#export\_graphviz(clf, out\_file=dot\_data, filled=True, rounded=True, special\_characters=True, feature\_names=features\_cols,*

*class\_names=['0','1'])*

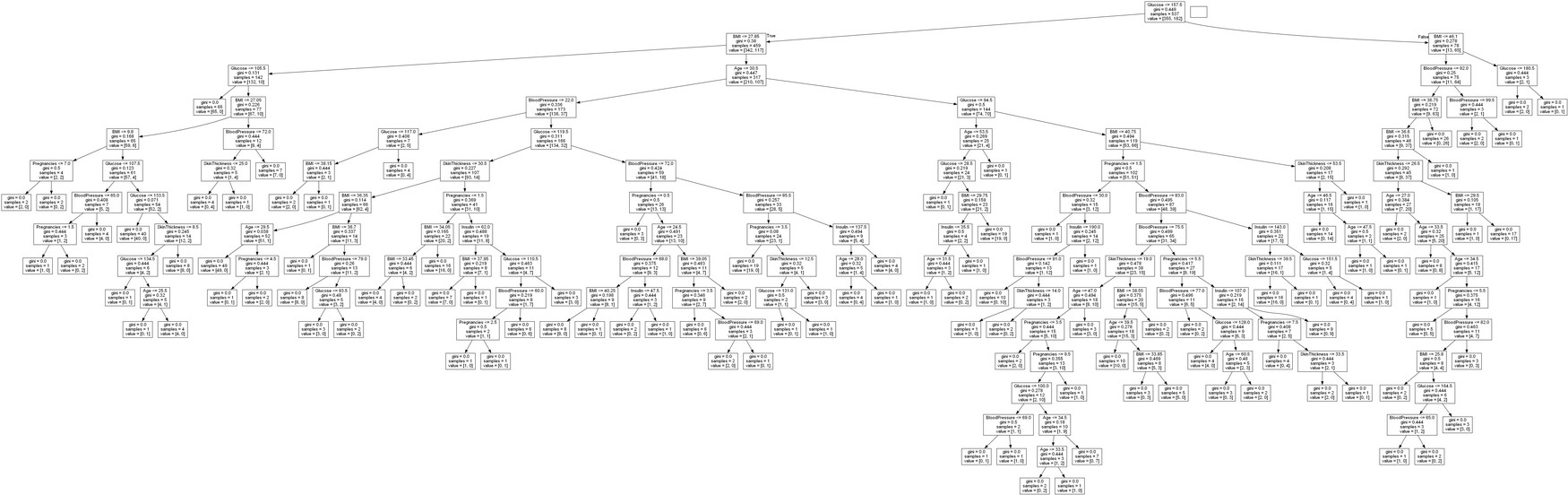
data1= export\_graphviz(clf, out\_file=None, feature\_names=feature\_cols) graph = pydotplus.graph\_from\_dot\_data(data1)

*#graph = pydotplus.graph\_from\_dot\_data(data1)*

graph.write\_png('outcome.png')

*#Image(graph.create\_png())*

Image(graph.create\_png())



Conclusion:- In conclusion, this assignment demonstrates the importance of decision trees in classification and regression. By understanding the theory behind decision trees and how to apply them to a dataset.