Data Science Intern @ LGM Virtual Intrnship 2021 (October) Intermediate Level Task Task 2: Prediction using Decision Tree Algorithm In []: **Import Libraries** In [18]: import numpy as np import pandas as pd import matplotlib.pyplot as plt import seaborn as sns %matplotlib inline from sklearn.model_selection import train_test_split from sklearn.tree import DecisionTreeClassifier from sklearn import metrics import graphviz from sklearn import tree **Import Dataset** In [19]: data=pd.read_csv("C:\\Users\\Admin\\Desktop\\Iris.csv", encoding=("ISO-8859-1"), low_memory=False) data.head(5) Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm Out[19]: Species 0 1 5.1 3.5 1.4 0.2 Iris-setosa 4.9 3.0 1.4 0.2 Iris-setosa **2** 3 4.7 3.2 1.3 0.2 Iris-setosa 4.6 1.5 0.2 Iris-setosa **4** 5 3.6 1.4 0.2 Iris-setosa 5.0 In [20]: data.shape Out[20]: (150, 6) data["Species"].value_counts() Iris-setosa Out[21]: Iris-versicolor 50 50 Iris-virginica Name: Species, dtype: int64 In [22]: data.isnull().sum() Ιd 0 Out[22]: SepalLengthCm 0 SepalWidthCm 0 PetalLengthCm 0 PetalWidthCm 0 Species dtype: int64 **Sepal Dimensions** plt.figure(figsize=(13,7)) sns.scatterplot(x=data["SepalLengthCm"],y=data["SepalWidthCm"],hue=data["Species"],s=100)Out[23]: <AxesSubplot:xlabel='SepalLengthCm', ylabel='SepalWidthCm'> Species Iris-setosa Iris-versicolor Iris-virginica 4.0 3.5 SepalWidthCm 0.6 2.5 2.0 5.5 4.5 5.0 6.0 6.5 7.0 7.5 8.0 SepalLengthCm In [24]: fig, axes=plt.subplots(figsize=(6,4)) plt.title("Distribution of Sepal Length") plt.hist(data["SepalLengthCm"]) fig, axes=plt.subplots(figsize=(6,4)) plt.title("Distribution of Sepal Width") plt.hist(data["SepalWidthCm"]) Out[24]: (array([4., 7., 22., 24., 38., 31., 9., 11., 2., 2.]), array([2. , 2.24, 2.48, 2.72, 2.96, 3.2 , 3.44, 3.68, 3.92, 4.16, 4.4]), <BarContainer object of 10 artists>) Distribution of Sepal Length 25 20 15 10 5 4.5 5.5 6.0 6.5 7.0 7.5 Distribution of Sepal Width 35 30 25 20 15 10 2.5 2.0 **Petal Dimensions** In [25]: plt.figure(figsize=(13,7)) sns.scatterplot(x=data["PetalLengthCm"], y=data["PetalWidthCm"], hue=data["Species"], s=100) Out[25]: <AxesSubplot:xlabel='PetalLengthCm', ylabel='PetalWidthCm'> Species Iris-setosa Iris-versicolor Iris-virginica 2.0 PetalWidthCm 1.0 0.5 0.0 PetalLengthCm In [26]: fig, axes=plt.subplots(figsize=(6,4)) plt.title("Distribution of Petal Length") plt.hist(data["PetalLengthCm"]) fig, axes=plt.subplots(figsize=(6,4)) plt.title("Distribution of Petal Width") plt.hist(data["PetalWidthCm"]) Out[26]: (array([41., 8., 1., 7., 8., 33., 6., 23., 9., 14.]), array([0.1, 0.34, 0.58, 0.82, 1.06, 1.3, 1.54, 1.78, 2.02, 2.26, 2.5]), <BarContainer object of 10 artists>) Distribution of Petal Length 35 30 25 20 15 10 5 Distribution of Petal Width 40 35 30 25 20 15 10 0.5 1.0 1.5 2.0 2.5 Correlation between Dependent and Independent variables In [27]: data1=data.drop("Id", axis=1) data1.head() data1["Species"].replace("Iris-setosa", "0", inplace=True) data1["Species"].replace("Iris-versicolor", "1", inplace=True) data1["Species"].replace("Iris-virginica", "2", inplace=True) data1["Species"]=data1.Species.astype(int) data1.head() Out[27]: SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm Species 0 5.1 3.5 1.4 0.2 0 4.9 3.0 1.4 0.2 0 2 4.7 3.2 1.3 0.2 0 4.6 3.1 1.5 0.2 0 5.0 0.2 3.6 1.4 0 In [28]: plt.figure(figsize=(8,4)) sns.heatmap(data1.corr(), annot=True) Out[28]: <AxesSubplot:> - 1.0 0.87 SepalLengthCm · -0.11 0.82 0.78 - 0.8 - 0.6 -0.11 1 -0.42 -0.36 -0.42 SepalWidthCm -- 0.4 PetalLengthCm · 0.87 -0.42 1 0.96 0.95 0.2 -0.36 0.82 0.96 1 0.96 PetalWidthCm · 0.0 -0.2-0.42 0.95 0.78 0.96 1 Species SepalWidthCm PetalLengthCm PetalWidthCm **Model Building** In [29]: train, test=train_test_split(data1, test_size=0.3) In [30]: train.shape, test.shape Out[30]: ((105, 5), (45, 5)) In [31]: train_x=train[["SepalLengthCm", "SepalWidthCm", "PetalLengthCm", "PetalWidthCm"]] train_y=train.Species test_x=test[["SepalLengthCm", "SepalWidthCm", "PetalLengthCm", "PetalWidthCm"]] test_y=test.Species In [32]: dtree=DecisionTreeClassifier() dtree.fit(train_x, train_y) predictions=dtree.predict(test_x) accuracy=metrics.accuracy_score(predictions, test_y) accuracy Out[32]: 0.977777777777777 In [33]: x=data1[["SepalLengthCm", "SepalWidthCm", "PetalLengthCm", "PetalWidthCm"]] y=data1.Species dtree1=DecisionTreeClassifier() dtree1.fit(x,y) Out[33]: DecisionTreeClassifier() Visualization In [34]: fig = plt.figure(figsize=(25,20)) tree.plot_tree(dtree1, feature_names=["SepalLengthCm", "SepalWidthCm", "PetalLengthCm", "PetalWidthCm"], class_names=["0", "1", "2"], Out[34]: [Text(697.5, 996.6, 'PetalLengthCm <= 2.45\ngini = 0.667\nsamples = 150\nvalue = [50, 50, 50]\nclass = 0'), Text(590.1923076923077, 815.40000000000001, 'gini = 0.0\nsamples = 50\nvalue = [50, 0, 0]\nclass = 0'),
Text(804.8076923076923, 815.4000000000001, 'PetalWidthCm <= 1.75\ngini = 0.5\nsamples = 100\nvalue = [0, 50, 50]\nclass = 1'), $Text(429.2307692307692, 634.2, 'PetalLengthCm <= 4.95 \ngini = 0.168 \nsamples = 54 \nvalue = [0, 49, 5] \nclass = 1'),$ Text(214.6153846153846, 453.0, 'PetalWidthCm <= 1.65\ngini = 0.041\nsamples = 48\nvalue = [0, 47, 1]\nclass = 1'), $\text{Text}(107.3076923076923, 271.7999999999999995, 'gini = 0.0 \nsamples = 47 \nvalue = [0, 47, 0] \nclass = 1'), \\ \text{Text}(321.9230769230769, 271.7999999999995, 'gini = 0.0 \nsamples = 1 \nvalue = [0, 0, 1] \nclass = 2'),$ Text(643.8461538461538, 453.0, 'PetalWidthCm <= 1.55\ngini = 0.444\nsamples = 6\nvalue = [0, 0, 1]\nclass = 2'),

Text(536.5384615384615, 271.7999999999995, 'gini = 0.0\nsamples = 3\nvalue = [0, 0, 3]\nclass = 2'),

Text(751.1538461538462, 271.7999999999995, 'PetalLengthCm <= 5.45\ngini = 0.444\nsamples = 3\nvalue = [0, 2, 1]\nclass = 1'),

Text(643.8461538461538, 90.5999999999991, 'gini = 0.0\nsamples = 2\nvalue = [0, 2, 0]\nclass = 1'),

Text(858.4615384615385, 90.5999999999991, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]\nclass = 2'), Text(1180.3846153846155, 634.2, 'PetalLengthCm <= 4.85\ngini = 0.043\nsamples = 46\nvalue = [0, 1, 45]\nclass = 2'),

Text(1073.076923076923, 453.0, 'SepalWidthCm <= 3.1\ngini = 0.444\nsamples = 3\nvalue = [0, 1, 2]\nclass = 2'),

Text(965.7692307692307, 271.7999999999995, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2]\nclass = 2'),

Text(1180.3846153846155, 271.7999999999995, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]\nclass = 1'), Text(1287.6923076923076, 453.0, 'gini = 0.0\nsamples = 43\nvalue = [0, 0, 43]\nclass = 2')] PetalLengthCm <= 2.45 gini = 0.667samples = 150value = [50, 50, 50]class = 0PetalWidthCm \leq 1.75 gini = 0.0gini = 0.5samples = 50samples = 100value = [50, 0, 0]value = [0, 50, 50]class = 0class = 1PetalLengthCm <= 4.85 PetalLengthCm <= 4.95 gini = 0.168gini = 0.043samples = 54samples = 46value = [0, 49, 5]value = [0, 1, 45]class = 1class = 2

PetalWidthCm <= 1.65

gini = 0.041

samples = 48

value = [0, 47, 1]

class = 1

gini = 0.0

samples = 1

value = [0, 0, 1]

class = 2

gini = 0.0

samples = 47

value = [0, 47, 0]

class = 1

In []:

PetalWidthCm <= 1.55

gini = 0.444

samples = 6

value = [0, 2, 4]

class = 2

gini = 0.0

samples = 2

value = [0, 2, 0]

class = 1

gini = 0.0

samples = 3

value = [0, 0, 3]

class = 2

PetalLengthCm <= 5.45

gini = 0.444

samples = 3

value = [0, 2, 1]

class = 1

gini = 0.0

samples = 1

value = [0, 0, 1]

class = 2

SepalWidthCm <= 3.1

gini = 0.444

samples = 3

value = [0, 1, 2]

class = 2

gini = 0.0

samples = 1

value = [0, 1, 0]

class = 1

gini = 0.0

samples = 2

value = [0, 0, 2]

class = 2

gini = 0.0

samples = 43

value = [0, 0, 43]

class = 2

Tanvi Bhosle