

ML LAB

ASSIGNMENT 1

NAME: SWAPNIL GHOSH
ROLL: 001911001067
DEPT: IT
YEAR: 4TH

IRIS - CLASSIFICATION



DECISION TREE CLASSIFIER - CODE & EVALUATION

```
dtc > No Selection

1 import matplotlib.pyplot as plt
2 from sklearn.datasets import load_iris
3 from sklearn import tree
4 from sklearn.model_selection import train_test_split
5 from sklearn.tree import DecisionTreeClassifier
6 from sklearn.metrics import classification_report, confusion_matrix
7
8 iris = load_iris()
9 X, y = iris.data, iris.target
10 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
11                                                    random_state=0)
12 classifier = DecisionTreeClassifier(criterion="gini", max_depth=2,
13                                   splitter="best", max_features = 2, max_leaf_nodes=3, random_state=1)
14 model = classifier.fit(X_train, y_train)
15 y_pred = classifier.predict(X_test)
16 print("Confusion Matrix:")
17 print(confusion_matrix(y_test, y_pred))
18
19 print("-----")
20
21 print("Performance Evaluation:")
22 print(classification_report(y_test, y_pred))
23
24 tree.plot_tree(decision_tree = model, feature_names = iris.feature_names,
25               class_names = iris.target_names, filled=True, rounded=True)
26 plt.savefig("iris.png")
```

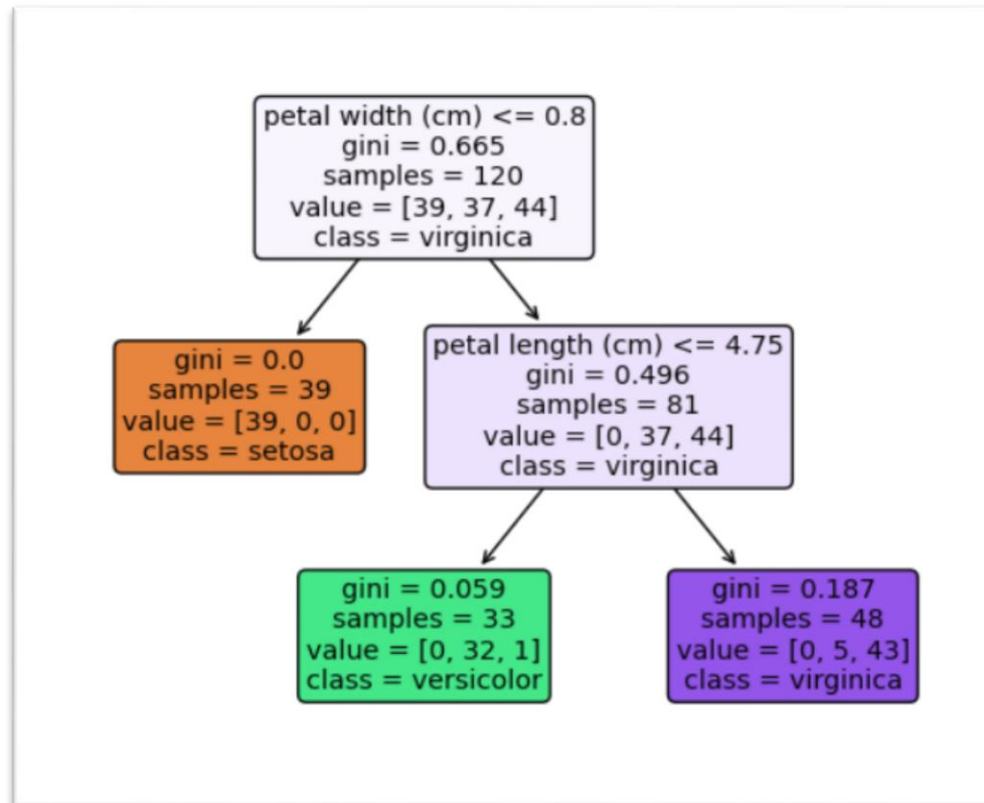
Confusion Matrix:

```
[[11  0  0]
 [ 0 12  1]
 [ 0  0  6]]
```

Performance Evaluation:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11
1	1.00	0.92	0.96	13
2	0.86	1.00	0.92	6
accuracy			0.97	30
macro avg	0.95	0.97	0.96	30
weighted avg	0.97	0.97	0.97	30

DECISION TREE CLASSIFIER - TREE PLOT



NAIVE BAYES - MULTINOMIALNB CLASSIFIER

```
1 from sklearn.datasets import load_iris
2 from sklearn.model_selection import train_test_split
3 from sklearn.naive_bayes import MultinomialNB
4 from sklearn.metrics import classification_report, confusion_matrix
5
6 iris = load_iris()
7 X, y = iris.data, iris.target
8 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
9                                                    random_state=0)
10 classifier = MultinomialNB(alpha=1.0, fit_prior=False)
11 classifier.fit(X_train, y_train)
12 y_pred = classifier.predict(X_test)
13
14 print("Confusion Matrix:")
15 print(confusion_matrix(y_test, y_pred))
16
17 print("-----")
18
19 print("Performance Evaluation:")
20 print(classification_report(y_test, y_pred))
21
```

CONFUSION MATRIX:

```
[11  0  0]
[ 0 13  0]
[ 0  0  6]
```

Performance Evaluation:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11
1	1.00	1.00	1.00	13
2	1.00	1.00	1.00	6
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

NAIVE BAYES - BERNOLLINB CLASSIFIER

```
1 from sklearn.datasets import load_iris
2 from sklearn.model_selection import train_test_split
3 from sklearn.naive_bayes import BernoulliNB
4 from sklearn.metrics import classification_report, confusion_matrix
5
6 iris = load_iris()
7 X, y = iris.data, iris.target
8 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
9                                                    random_state=1)
10
11 classifier = BernoulliNB(alpha=1.0, binarize=1.7, fit_prior=False)
12 classifier.fit(X_train, y_train)
13 y_pred = classifier.predict(X_test)
14 print("Confusion Matrix:")
15 print(confusion_matrix(y_test, y_pred))
16
17 print("-----")
18
19 print("Performance Evaluation:")
20 print(classification_report(y_test, y_pred))
21
```

Confusion Matrix:

```
[10  1  0]
[ 0 13  0]
[ 0  0  6]
```

Performance Evaluation:

	precision	recall	f1-score	support
0	1.00	0.91	0.95	11
1	0.93	1.00	0.96	13
2	1.00	1.00	1.00	6
accuracy			0.97	30
macro avg	0.98	0.97	0.97	30
weighted avg	0.97	0.97	0.97	30

DIABETES - REGRESSION



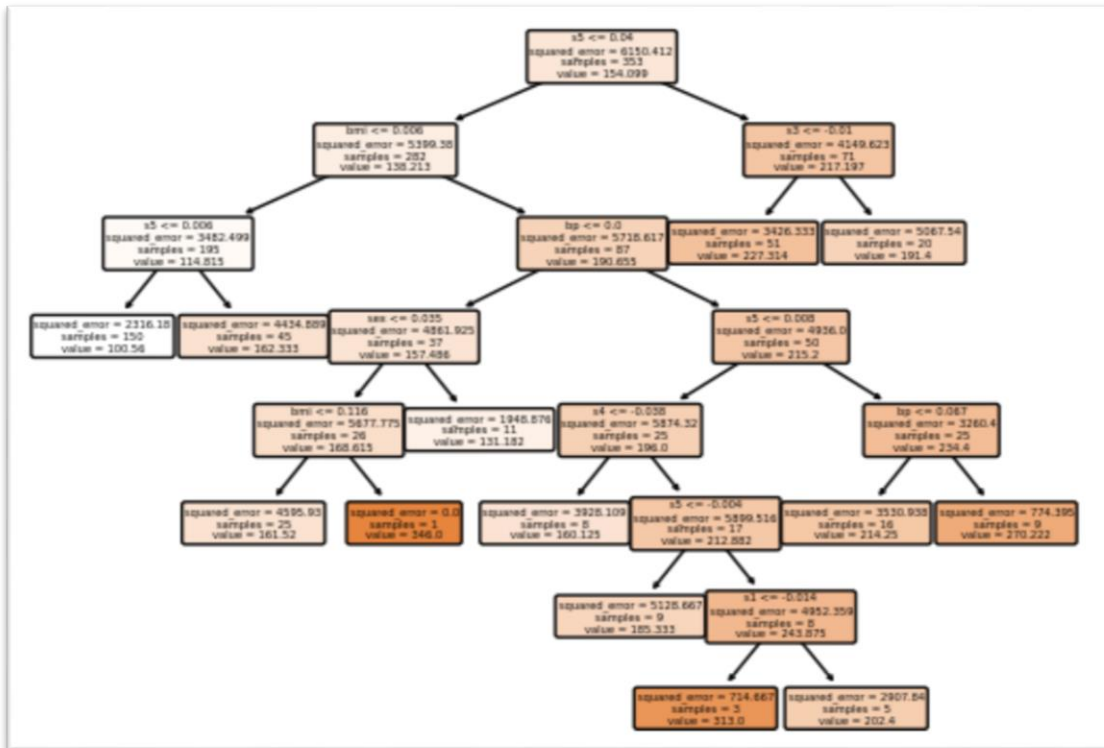
DECISION TREE REGRESSOR - CODE & EVALUATION

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from sklearn.datasets import load_diabetes
4 from sklearn import tree
5 from sklearn.model_selection import cross_val_score
6 from sklearn.model_selection import train_test_split
7 from sklearn.tree import DecisionTreeRegressor
8
9
10 diabetes = load_diabetes()
11 X, y = diabetes.data, diabetes.target
12 X_train, X_test, y_train, y_test = train_test_split(X, y,
13     test_size=0.2, random_state=7)
14 regressor = DecisionTreeRegressor(criterion="squared_error",
15     splitter = "random", max_features="sqrt", ccp_alpha=41,
16     random_state=46)
17 model = regressor.fit(X_train, y_train)
18 y_pred = regressor.predict(X_test)
19
20 print("Performance Evaluation:\n")
21
22 print("RMSE: ", np.sqrt(np.mean((y_pred - y_test)**2)))
23 print("MAE: ", np.mean(np.abs(y_pred - y_test)))
24 print("R2: ", regressor.score(X_test, y_test))
25
26 tree.plot_tree(regressor, feature_names = diabetes.feature_names,
27     filled=True, rounded=True)
28 plt.savefig("diabetes.png")
```

diabetes.py
Performance Evaluation:

RMSE: 51.79107344514936
MAE: 40.67246383197474
R2: 0.4612850990313666

DECISION TREE REGRESSOR - TREE PLOT



BREAST CANCER WISCONSIN - CLASSIFICATION



DECISION TREE CLASSIFIER - CODE & EVALUATION

```
1 import matplotlib.pyplot as plt
2 from sklearn.datasets import load_breast_cancer
3 from sklearn import tree
4 from sklearn.tree import DecisionTreeClassifier
5 from sklearn.metrics import classification_report, confusion_matrix
6 from sklearn.model_selection import train_test_split
7
8 cancer = load_breast_cancer()
9 X, y = cancer.data, cancer.target
10 X_train, X_test, y_train, y_test = train_test_split(X, y,
11                                                    test_size=0.2, random_state=0)
12
13 classifier = DecisionTreeClassifier(criterion="gini", max_depth=4,
14                                   splitter="best", max_features = "sqrt", max_leaf_nodes=8,
15                                   random_state=5)
16 model = classifier.fit(X_train, y_train)
17 y_pred = classifier.predict(X_test)
18
19 print("Confusion Matrix:")
20 print(confusion_matrix(y_test, y_pred))
21
22 print("-----")
23
24 print("Performance Evaluation:")
25 print(classification_report(y_test, y_pred))
26
27 tree.plot_tree(decision_tree = model, feature_names =
28               cancer.feature_names, class_names = cancer.target_names,
29               filled=True, rounded=True)
30 plt.savefig("breast_cancer_wisconsin.png")
```

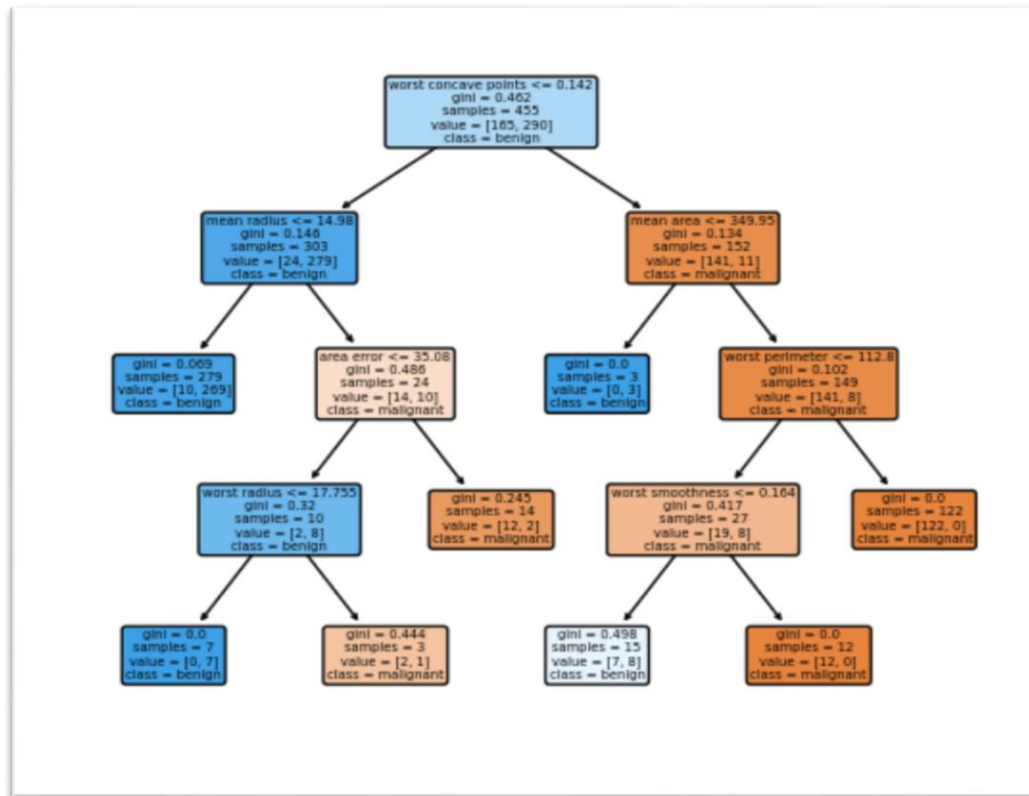
Confusion Matrix:

```
[[46  1]
 [ 1 66]]
```

Performance Evaluation:

	precision	recall	f1-score	support
0	0.98	0.98	0.98	47
1	0.99	0.99	0.99	67
accuracy			0.98	114
macro avg	0.98	0.98	0.98	114
weighted avg	0.98	0.98	0.98	114

DECISION TREE CLASSIFIER - TREE PLOT



NAIVE BAYES - MULTINOMIALNB CLASSIFIER

```
1 from sklearn.datasets import load_breast_cancer
2 from sklearn.naive_bayes import MultinomialNB
3 from sklearn.metrics import classification_report, confusion_matrix
4 from sklearn.model_selection import train_test_split
5
6 cancer = load_breast_cancer()
7 X, y = cancer.data, cancer.target
8 X_train, X_test, y_train, y_test = train_test_split(X, y,
9                                                     test_size=0.2)
10
11 classifier = MultinomialNB(alpha=0.5, fit_prior=True,
12                             class_prior=[0.45, 0.55])
13 classifier.fit(X_train, y_train)
14 y_pred = classifier.predict(X_test)
15
16 print("Confusion Matrix:")
17 print(confusion_matrix(y_test, y_pred))
18
19 print("-----")
20
21 print("Performance Evaluation:")
22 print(classification_report(y_test, y_pred))
23
```

Confusion Matrix:

```
[[34  4]
 [ 3 73]]
```

Performance Evaluation:

	precision	recall	f1-score	support
0	0.92	0.89	0.91	38
1	0.95	0.96	0.95	76
accuracy			0.94	114
macro avg	0.93	0.93	0.93	114
weighted avg	0.94	0.94	0.94	114

NAIVE BAYES - GAUSSIANNB CLASSIFIER

```
1 from sklearn.datasets import load_breast_cancer
2 from sklearn.naive_bayes import GaussianNB
3 from sklearn.metrics import classification_report, confusion_matrix
4 from sklearn.model_selection import train_test_split
5
6 cancer = load_breast_cancer()
7 X, y = cancer.data, cancer.target
8 X_train, X_test, y_train, y_test = train_test_split(X, y,
9                                                    test_size=0.2, random_state=0)
10
11 classifier = GaussianNB(var_smoothing=1e-8)
12 classifier.fit(X_train, y_train)
13 y_pred = classifier.predict(X_test)
14
15 print("Confusion Matrix:")
16 print(confusion_matrix(y_test, y_pred))
17
18 print("-----")
19
20 print("Performance Evaluation:")
21 print(classification_report(y_test, y_pred))
```

Confusion Matrix:

```
[[44  3]
 [ 2 65]]
```

Performance Evaluation:

	precision	recall	f1-score	support
0	0.96	0.94	0.95	47
1	0.96	0.97	0.96	67
accuracy			0.96	114
macro avg	0.96	0.95	0.95	114
weighted avg	0.96	0.96	0.96	114

NAIVE BAYES - BERNOLLI NB CLASSIFIER

```
1 from sklearn.datasets import load_breast_cancer
2 from sklearn.naive_bayes import BernoulliNB
3 from sklearn.metrics import classification_report,
  confusion_matrix
4 from sklearn.model_selection import train_test_split
5
6 cancer = load_breast_cancer()
7 X, y = cancer.data, cancer.target
8 X_train, X_test, y_train, y_test = train_test_split(X, y,
  test_size=0.2, random_state=0)
9
10 classifier = BernoulliNB(alpha=1.0,
  binarize=16.8, fit_prior=False)
11 classifier.fit(X_train, y_train)
12 y_pred = classifier.predict(X_test)
13
14 print("Confusion Matrix:")
15 print(confusion_matrix(y_test, y_pred))
16
17 print("-----")
18
19 print("Performance Evaluation:")
20 print(classification_report(y_test, y_pred))
21
```

Confusion Matrix:

```
[[42  5]
 [ 1 66]]
```

Performance Evaluation:

	precision	recall	f1-score	support
0	0.98	0.89	0.93	47
1	0.93	0.99	0.96	67
accuracy			0.95	114
macro avg	0.95	0.94	0.94	114
weighted avg	0.95	0.95	0.95	114