

In-Lab

Task 1

Importing Required libraries and dataset

```
!apt-get install graphviz
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.tree import export_graphviz
import graphviz

from google.colab import files
u = files.upload()
```

Task 2

```
col_names =
['pregnant','glucose','bp','skin','insulin','bmi','pedigree','age','label']
pima = pd.read_csv("diabetes.csv",header = None,names=col_names)

pima_df = pima.head()
print(pima_df)
```

	pregnant	glucose	bp	skin	insulin	bmi	pedigree	age	label
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Figure 1

Task 3

```
feature_cols =  
['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree']  
X = pima[feature_cols]  
y = pima.label
```

Task 4

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =  
0.3, random_state = 1)
```

Task 5

```
clf = DecisionTreeClassifier()  
clf = clf.fit(X_train, y_train)  
  
y_pred = clf.predict(X_test)
```

Task 6

```
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

Accuracy: 0.6666666666666666

Figure 2

Task 7

```
dot_data =  
export_graphviz(clf, out_file=None, feature_names=X_train.columns, class_names=[str(x)  
for x in clf.classes_], filled=True, rounded=True, special_characters=True)  
graph = graphviz.Source(dot_data)  
graph.render("decision_tree")  
graph.view("decision_tree")
```

Task 8

```
clf = DecisionTreeClassifier(criterion = "entropy", max_depth=3)
clf = clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)

print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

Accuracy: 0.7705627705627706

Figure 3

Task 9

```
dot_data =
export_graphviz(clf,out_file=None,feature_names=X_train.columns,class_n
ames=[str(x) for x in clf.classes_],filled=True, rounded=True,
special_characters=True)
graph = graphviz.Source(dot_data)
graph.render("decision_tree")
graph.view("decision_tree")
```