```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classification
from sklearn.metrics import accuracy_score,confusion_matrix,f1_score
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from matplotlib import rcParams
from sklearn.tree import DecisionTreeClassifier
```

```
data=pd.read_csv("/content/pima(For Random Forest Program).csv")
data
```

```
data.columns
```

```
x=data.drop("Outcome",axis=1)
x
```

```
y=data["Outcome"]
y
```

+ Code + Text

```
scaler=StandardScaler()
x_scaled=scaler.fit_transform(x)
```

```
X_train,X_test,Y_train,Y_test=train_test_split(x_scaled,y,stratify=y,test_size=0.10,random_state=34)
```

```
classifier=RandomForestClassifier(n_estimators=100)
classifier.fit(X_train,Y_train)
```

```
Y_pred=classifier.predict(X_test)
print("Accuracy : ",accuracy_score(Y_test,Y_pred))

feature_importances_df = pd.DataFrame({"feature":list(x.columns),"importance":classifier.feature_importances_}).sort_values("importance",

feature_importances_df
```

```
confusion_matrix(Y_test,Y_pred)
```

```
print("f1_score : ",f1_score(Y_test,Y_pred))
```

```
clf=DecisionTreeClassifier()
clf.fit(X_train,Y_train)
Y_pred=clf.predict(X_test)

print("Accuracy - DecisionTree : ",accuracy_score(Y_test,Y_pred))
print("f1_score : ",f1_score(Y_test,Y_pred))
```

✓ 0s completed at 11:07 AM ● ×