In [8]:

```
import pandas as pd


df = pd.read_csv("Tensile modulus.csv" , encoding='unicode_escape')
print("The Dataset is: ")
df
```

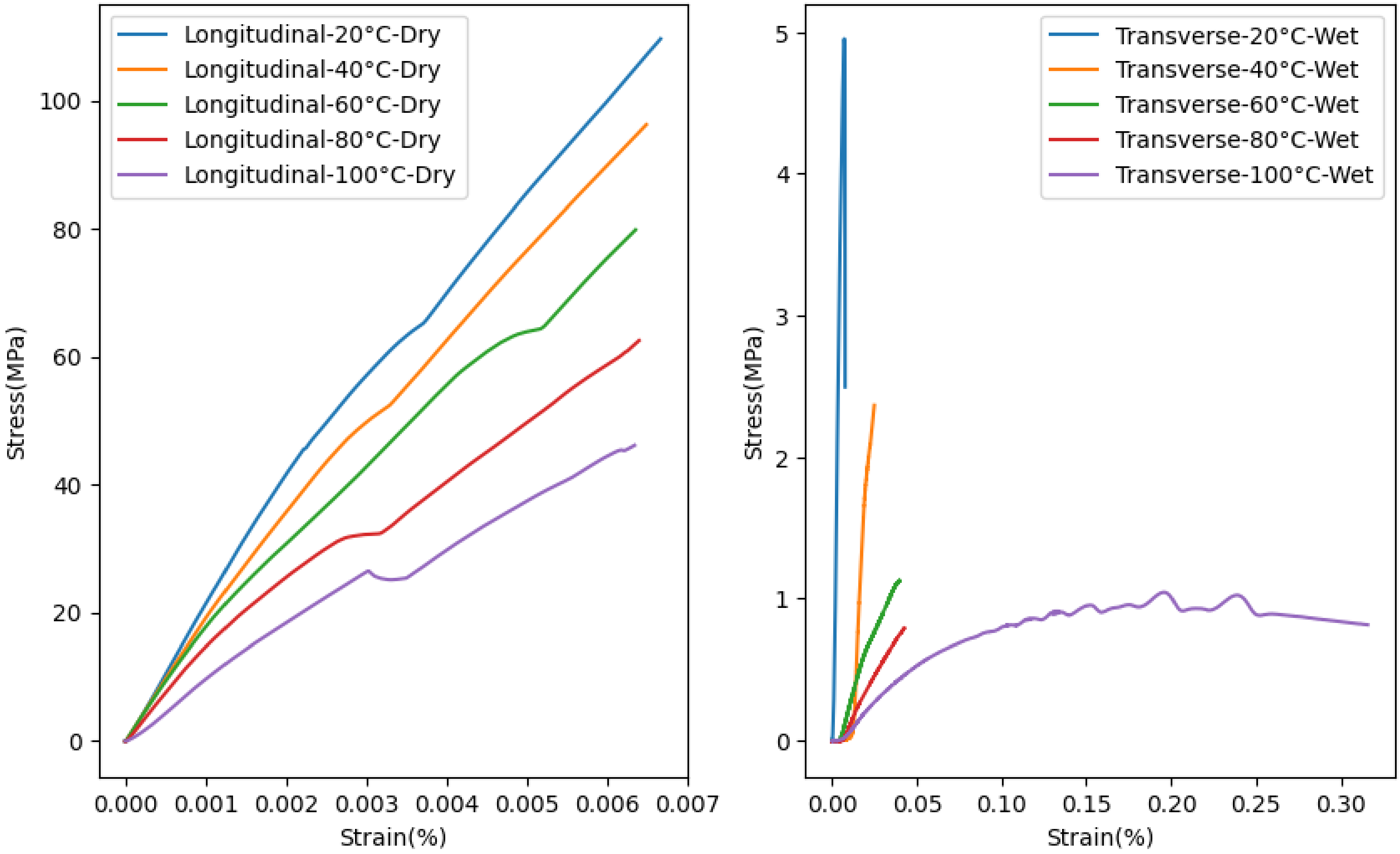
```
The Dataset is:
```

Out[8]:

	Specimen Index	direction	temperature (°C)	Humidity	Tensile modulus (GPa)
0	1	longitude	20	50%	16.302
1	2	longitude	20	50%	15.065
2	3	longitude	20	50%	13.124
3	4	longitude	20	50%	12.690
4	5	longitude	20	50%	11.714
...	...	...	...	...	...
95	96	transverse	100	Immersed	0.020
96	97	transverse	100	Immersed	0.021
97	98	transverse	100	Immersed	0.022
98	99	transverse	100	Immersed	0.016

In [44]:

```
import matplotlib.pyplot as plt

df1 = pd.read_csv("L-20-dry-1.csv")
df2 = pd.read_csv("L-40-dry-1.csv")
df3 = pd.read_csv("L-60-dry-1.csv")
df4 = pd.read_csv("L-80-dry-1.csv")
df5 = pd.read_csv("L-100-dry-1.csv")

df6 = pd.read_csv("W-20-wet-1.csv")
df7 = pd.read_csv("W-40-wet-2.csv")
df8 = pd.read_csv("W-60-wet-1.csv")
df9 = pd.read_csv("W-80-wet-1.csv")
df10 = pd.read_csv("W-100-wet-1.csv")

y1 = [df1["Stress (MPa)"] , df2["Stress (MPa)"] , df3["Stress (MPa)"] , df4["Stress (MPa)"] , df5["Stres
x1 = [df1["Strain"] , df2["Strain"] , df3["Strain"] , df4["Strain"] , df5["Strain"]]

y2 = [df6["Stress (MPa)"] , df7["Stress (MPa)"] , df8["Stress (MPa)"] , df9["Stress (MPa)"] , df10["Stre
x2 = [df6["Strain"] , df7["Strain"] , df8["Strain"] , df9["Strain"] , df10["Strain"]]

plt.figure(figsize=(10,6))

plt.subplot(1 , 2 , 1)
for i in range(5):
    plt.plot(x1[i] , y1[i] , label = f"Longitudinal-{(i+1) * 20}°C-Dry")
plt.xlabel("Strain(%)")
plt.ylabel("Stress(MPa)")
plt.legend()

plt.subplot(1 , 2 , 2)
for i in range(5):
    plt.plot(x2[i] , y2[i] , label = f"Transverse-{(i+1) * 20}°C-Wet")
plt.xlabel("Strain(%)")
plt.ylabel("Stress(MPa)")
plt.legend()

plt.show()
```


Longitudinal-20°C-Dry
Longitudinal-40°C-Dry
Longitudinal-60°C-Dry
Longitudinal-80°C-Dry
Longitudinal-100°C-Dry
Stress(MPa)
0
20
40
60
80
100
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0.007
Strain(%)
Transverse-20°C-Wet
Transverse-40°C-Wet
Transverse-60°C-Wet
Transverse-80°C-Wet
Transverse-100°C-Wet
Stress(MPa)
0
1
2
3
4
5
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Strain(%)

In [54]:

```
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
from sklearn.inspection import permutation_importance
import pandas as pd

df = pd.read_csv("Tensile modulus(normal).csv" , encoding='unicode_escape')
features = ["direction" , "temperature" , "Humidity"]
X = df[features]
y = df['Tensile modulus']

def DT(X , y , testData):
    output = []
    reg = DecisionTreeRegressor(random_state=0)
    reg.fit(X.values , y.values)

    predictedValue = reg.predict(testData)
    output.append(predictedValue[0])

    r2 = r2_score(y , reg.predict(X.values))
    output.append(r2)

    importance = reg.feature_importances_
    for i in range(3):
        output.append(importance[i])

    return output

def SVM(X , y , testData):
    output = []
    reg = SVR(kernel="rbf" , C=1 , epsilon=0.1)
    reg.fit(X.values , y.values)

    predictedValue = reg.predict(testData)
    output.append(predictedValue[0])

    r2 = r2_score(y , reg.predict(X.values))
    output.append(r2)


    importance = permutation_importance(reg , X.values , y.values)
    for i in range(3):
        output.append(importance['importances_mean'][i] / sum(importance['importances_mean']))

    return output

def RF(X , y , testData):
    output = []
    reg = RandomForestRegressor(random_state=0 , n_estimators=100)
    reg.fit(X.values , y.values)

    predictedValue = reg.predict(testData)
    output.append(predictedValue[0])

    r2 = r2_score(y , reg.predict(X.values))
    output.append(r2)


    importance = reg.feature_importances_
    for i in range(3):
        output.append(importance[i])

    return output
```

In [84]:

```
from IPython.display import display
test = [[0 , 20 , 50]]
df1 = pd.DataFrame({"Predicted Tensile Modulus" : [DT(X , y , test)[0] , SVM(X , y , test)[0] , RF(X , y
                  "Coefficient of Determination(R squared)" : [DT(X , y , test)[1] , SVM(X , y , test)[
df1.index = ["Decision Tree" , "Support Vector Machine" , "Random Forest"]

df2 = pd.DataFrame({"Decision Tree" : [DT(X , y , test)[2] , DT(X , y , test)[3] , DT(X , y , test)[4]]
                   "Support Vector Machine" : [SVM(X , y , test)[2] , SVM(X , y , test)[3] , SVM(X , y
                   "Random Forest" : [RF(X , y , test)[2] , RF(X , y , test)[3] , RF(X , y , test)[4]]})
df2.index = ["Fiber Orientation" , "Temperature" , "Humidity"]
df2.index.name = "Feature Importance"
display(df1)
print("-"*50)
display(df2)
```

	Predicted Tensile Modulus	**Coefficient of Determination(R squared)**
Decision Tree	13.779000	0.969057
Support Vector Machine	11.409722	0.942183
Random Forest	14.001540	0.968772

```
--------------------------------------------------
```

	Decision Tree	**Support Vector Machine**	**Random Forest**
Feature Importance			
Fiber Orientation	0.607056	0.659499	0.598331
Temperature	0.163298	0.151948	0.168610
Humidity	0.229646	0.204031	0.233059

In [ ]: