

## In lab Task 1:

### Code output:

```
model2 = tree.DecisionTreeRegressor()
model2.fit(X_train, y_train)
print("Decision Tree")
print("=====")

y_pred_train2 = model2.predict(X_train)
RMSE_train2 = mean_squared_error(y_train, y_pred_train2)
print("Decision Tree Train set: RMSE {}".format(RMSE_train2))

y_pred2 = model2.predict(X_test)
RMSE_test2 = mean_squared_error(y_test, y_pred2)
print("Decision Tree Test set: RMSE {}".format(RMSE_test2))
print("=====")

x_values = np.arange(len(y_test))
plt.scatter(x_values, y_test, color = 'red', label = 'Actual')
plt.scatter(x_values, y_pred2, color = 'green', label = 'Predicted')
plt.xlabel('Index or Sequence of values')
plt.ylabel('Values')
plt.title('Decision Tree Regression Test Set')
plt.legend()
plt.show()

x_values = np.arange(len(y_train))
plt.scatter(x_values, y_train, color = 'red', label = 'Actual')
plt.scatter(x_values, y_pred_train2, color = 'green', label = 'Predicted')
plt.xlabel('Index or Sequence of values')
plt.ylabel('Values')
plt.title('Decision Tree Regression Training Set')
plt.legend()
plt.show()
```

## In lab Task 2:

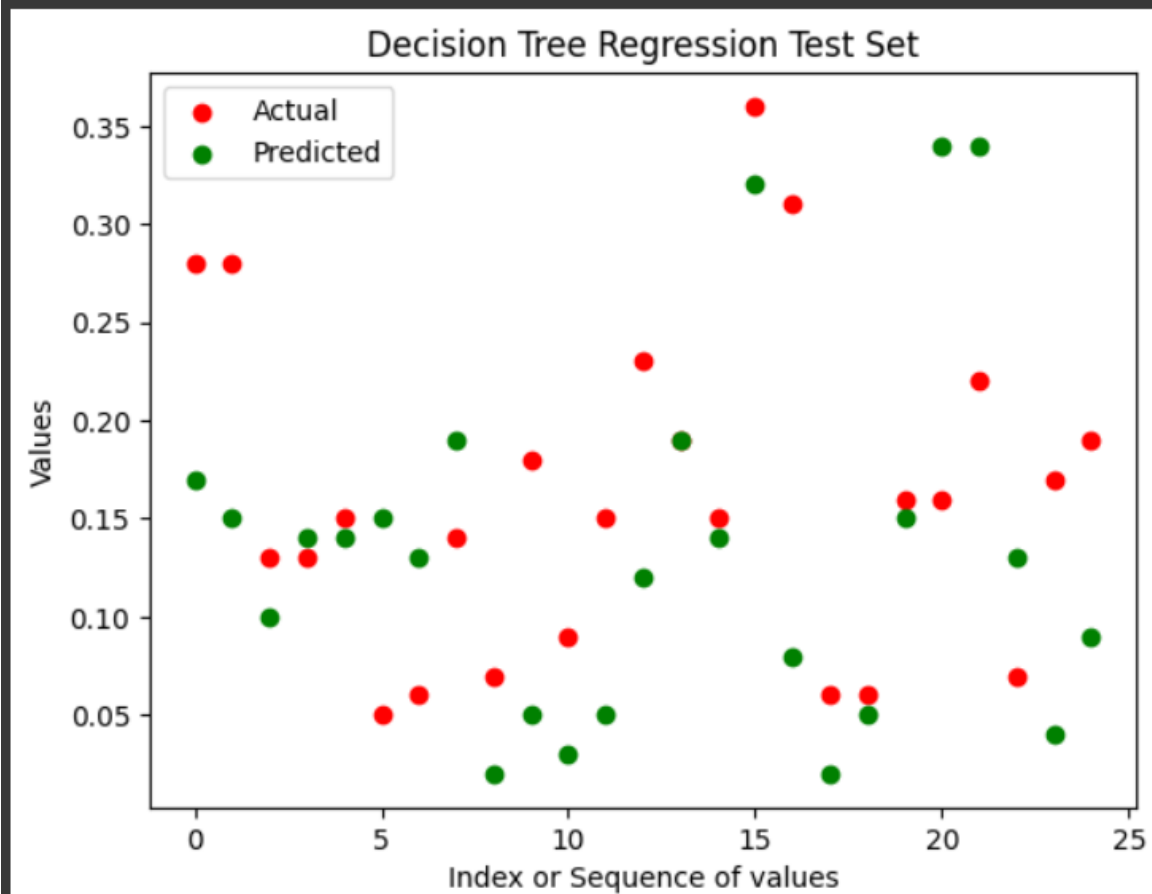
## Decision Tree

=====

Decision Tree Train set: RMSE 1.4739259778473743e-36

Decision Tree Test set: RMSE 0.009052

=====



## In lab Task 3:



### In lab Task 4:

```
model3 = RandomForestRegressor()
model3.fit(X_train, y_train)
print("Random Forest Regressor")
print("=====")

y_pred_train3 = model3.predict(X_train)
RMSE_train3 = mean_squared_error(y_train, y_pred_train3)
print("Random Forest Regressor Train set: RMSE {}".format(RMSE_train3))

y_pred3 = model3.predict(X_test)
RMSE_test3 = mean_squared_error(y_test, y_pred3)
print("Random Forest Regressor Test set: RMSE {}".format(RMSE_test3))
print("=====")

x_values = np.arange(len(y_test))
plt.scatter(x_values, y_test, color = 'red', label = 'Actual')
plt.scatter(x_values, y_pred3, color = 'green', label = 'Predicted')
plt.xlabel('Index or Sequence of values')
plt.ylabel('Values')
plt.title('Random Forest Regressor Test Set')
plt.legend()
plt.show()

x_values = np.arange(len(y_train))
plt.scatter(x_values, y_train, color = 'red', label = 'Actual')
plt.scatter(x_values, y_pred_train3, color = 'green', label = 'Predicted')
plt.xlabel('Index or Sequence of values')
plt.ylabel('Values')
plt.title('Random Forest Regressor Training Set')
plt.legend()
plt.show()
```

## In lab Task 5:

