```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_squared_error
from xgboost import XGBRegressor
from tqdm import tqdm


file_path = r'.xlsx'
data = pd.read_excel(file_path)


y = data.iloc[:, 0].values
X = data.iloc[:, 1:].values


xgb_params = {
    'n_estimators': 20,
    'max_depth': 1,
    'learning_rate': 0.1,
    'random_state': 50
}


xgb_model = XGBRegressor(**xgb_params)


num_random_states = 1000

results_df = pd.DataFrame(columns=['Random_State', 'Train_R2', 'Test_R2', 'Train_RMSE', 'Test_RMSE'])


for random_state in tqdm(range(1, num_random_states + 1)):

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=random_state)
    xgb_model.fit(X_train, y_train)

    y_train_pred = xgb_model.predict(X_train)
    y_test_pred = xgb_model.predict(X_test)

    train_r2 = r2_score(y_train, y_train_pred)
    test_r2 = r2_score(y_test, y_test_pred)
    train_rmse = np.sqrt(mean_squared_error(y_train, y_train_pred))
    test_rmse = np.sqrt(mean_squared_error(y_test, y_test_pred))
    results_df = pd.concat([results_df, pd.DataFrame({'Random_State': [random_state], 'Train_R2': [train_r2], 'Test_R2': [test_r2], 'Train_RMSE': [train_rmse], 'Test_RMSE': [test_rmse]})], ignore_index=True)

output_path = r'.xlsx'
results_df.to_excel(output_path, index=False)

print(f'Results saved to {output_path}')
```