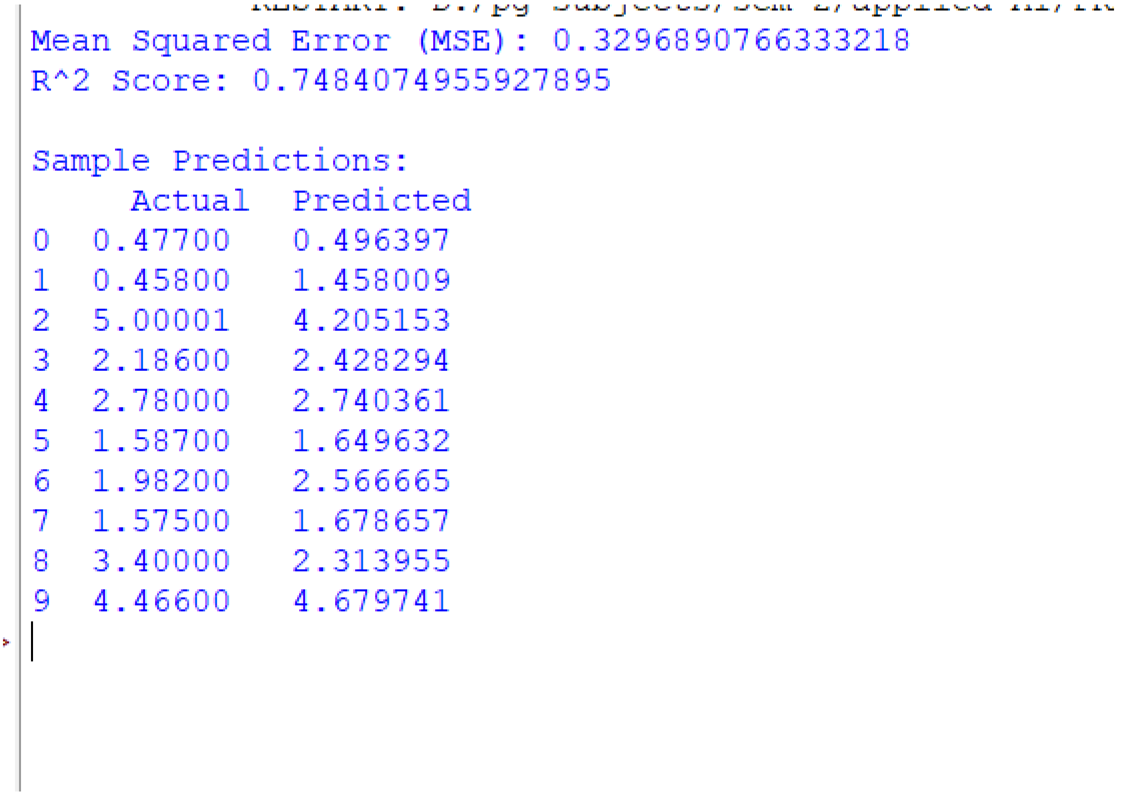
APPLIED ARTIFICIAL INTELLIGENCE

EXPERIMENT – 08

```
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, r2_score
import pandas as pd

# Load dataset
data = fetch_california_housing()
X = data.data
y = data.target
feature_names = data.feature_names

# Split and scale data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
scaler_X = StandardScaler()
scaler_y = StandardScaler()

X_train_scaled = scaler_X.fit_transform(X_train)
X_test_scaled = scaler_X.transform(X_test)
y_train_scaled = scaler_y.fit_transform(y_train.reshape(-1, 1)).ravel()

# Train SVR model
svr = SVR(kernel='rbf', C=10, gamma=0.1)
svr.fit(X_train_scaled, y_train_scaled)

# Predict and inverse transform
y_pred_scaled = svr.predict(X_test_scaled)
y_pred = scaler_y.inverse_transform(y_pred_scaled.reshape(-1, 1)).ravel()
```

```
# Evaluation

mse = mean_squared_error(y_test, y_pred)

r2 = r2_score(y_test, y_pred)


# Display results

print("Mean Squared Error (MSE):", mse)

print("R^2 Score:", r2)


# Optional: Compare actual vs predicted

comparison = pd.DataFrame({'Actual': y_test[:10], 'Predicted': y_pred[:10]})

print("\nSample Predictions:\n", comparison)
```

output:



```
Mean Squared Error (MSE): 0.3296890766333218
R^2 Score: 0.7484074955927895

Sample Predictions:
     Actual  Predicted
0  0.47700   0.496397
1  0.45800   1.458009
2  5.00001   4.205153
3  2.18600   2.428294
4  2.78000   2.740361
5  1.58700   1.649632
6  1.98200   2.566665
7  1.57500   1.678657
8  3.40000   2.313955
9  4.46600   4.679741
```