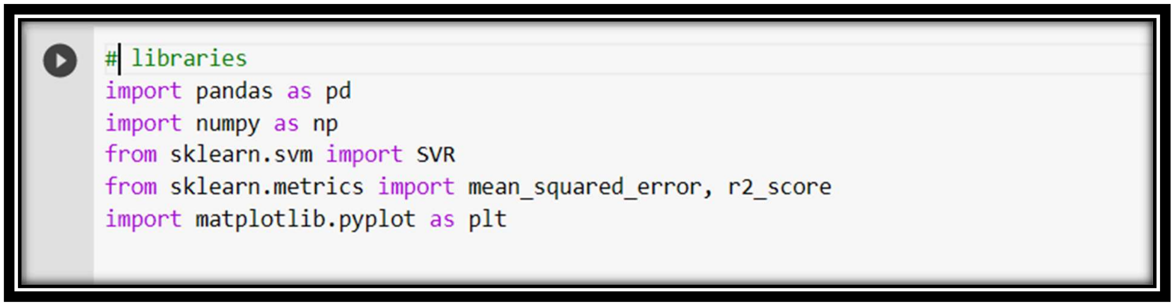
Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Computer Science and Engineering (Data Science)

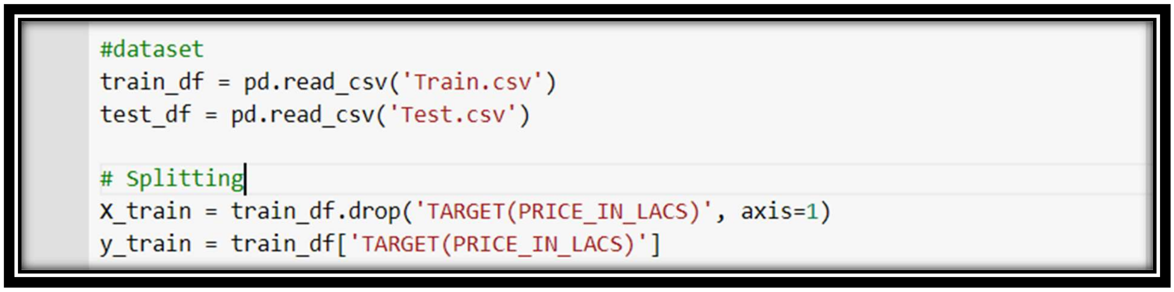
MACHINE LEARNING MINIPROJECT

NAME: DIVYESH KHUNT **SAPID :60009210116**

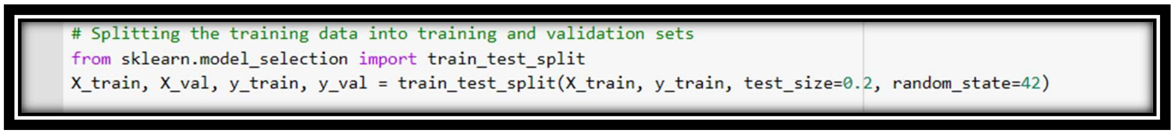
HOUSE PRICE PREDICTION



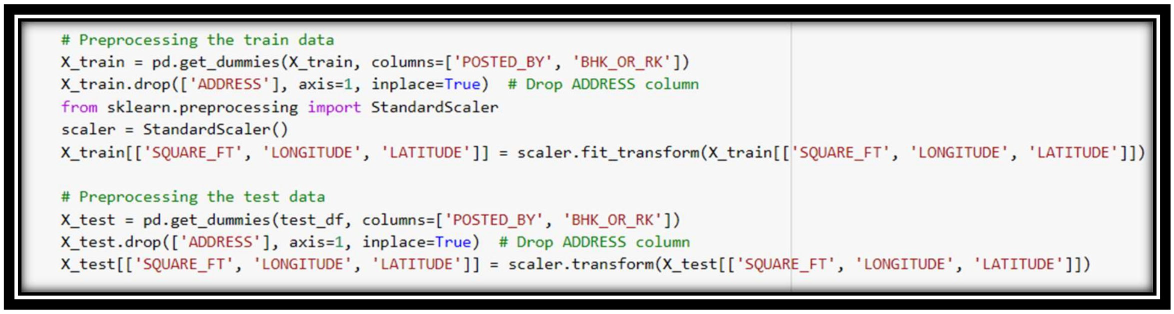
```
# libraries
import pandas as pd
import numpy as np
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
```



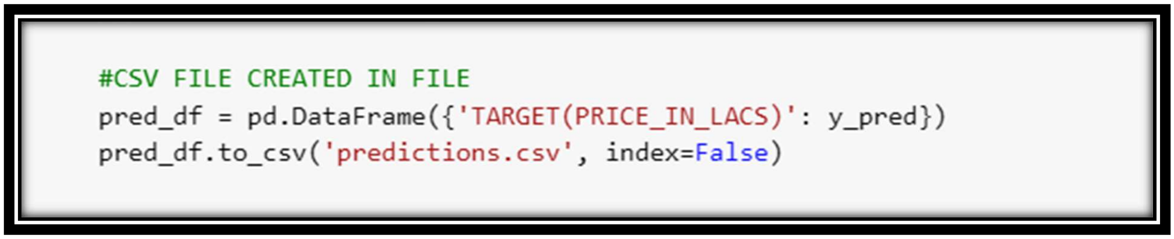
```
#dataset
train_df = pd.read_csv('Train.csv')
test_df = pd.read_csv('Test.csv')

# Splitting
X_train = train_df.drop('TARGET(PRICE_IN_LACS)', axis=1)
y_train = train_df['TARGET(PRICE_IN_LACS)']
```



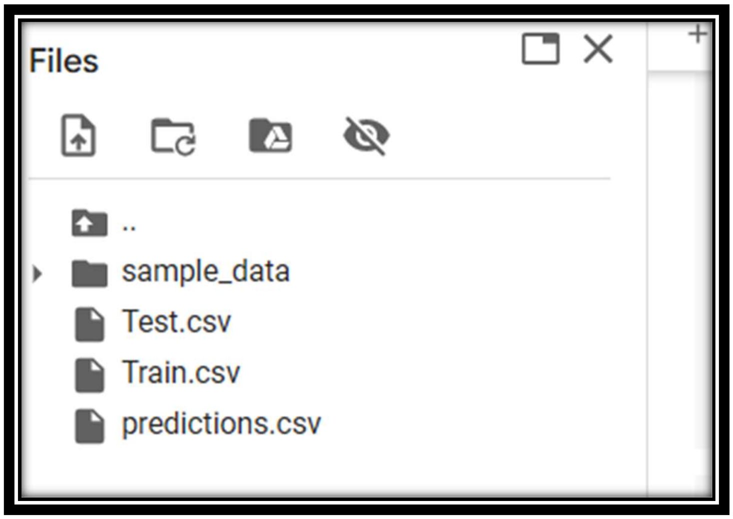
```
# Splitting the training data into training and validation sets
from sklearn.model_selection import train_test_split
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.2, random_state=42)
```



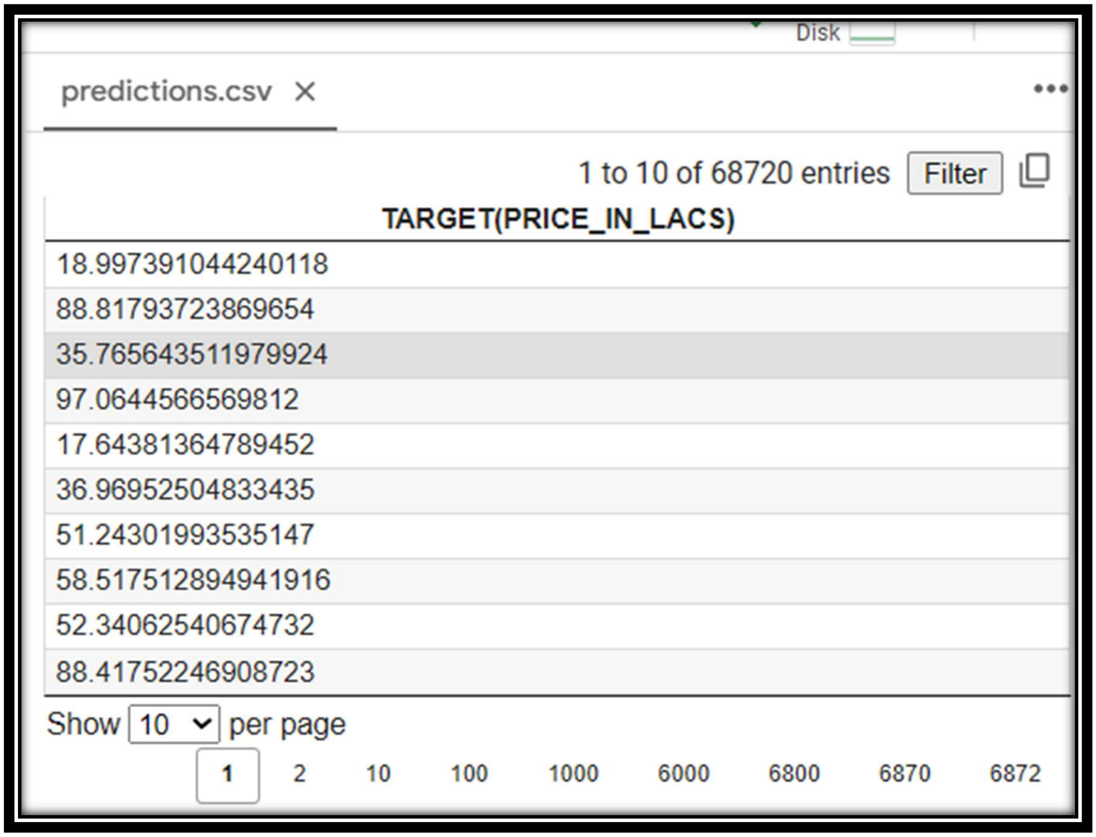
```
# Preprocessing the train data
X_train = pd.get_dummies(X_train, columns=['POSTED_BY', 'BHK_OR_RK'])
X_train.drop(['ADDRESS'], axis=1, inplace=True)  # Drop ADDRESS column
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train[['SQUARE_FT', 'LONGITUDE', 'LATITUDE']] = scaler.fit_transform(X_train[['SQUARE_FT', 'LONGITUDE', 'LATITUDE']])

# Preprocessing the test data
X_test = pd.get_dummies(test_df, columns=['POSTED_BY', 'BHK_OR_RK'])
X_test.drop(['ADDRESS'], axis=1, inplace=True)  # Drop ADDRESS column
X_test[['SQUARE_FT', 'LONGITUDE', 'LATITUDE']] = scaler.transform(X_test[['SQUARE_FT', 'LONGITUDE', 'LATITUDE']])
```



```
#CSV FILE CREATED IN FILE
pred_df = pd.DataFrame({'TARGET(PRICE_IN_LACS)': y_pred})
pred_df.to_csv('predictions.csv', index=False)
```

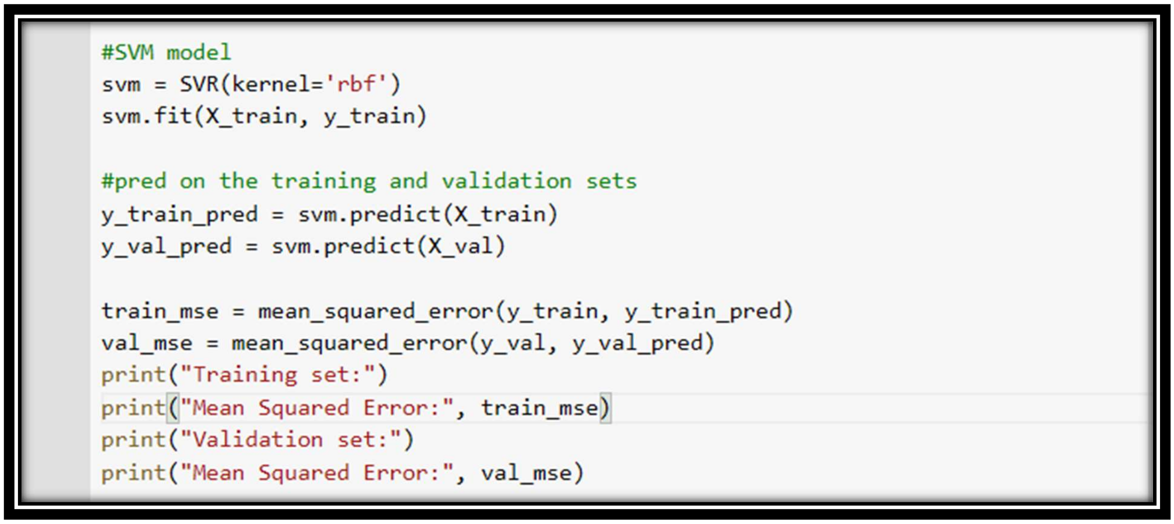


..
sample_data
Test.csv
Train.csv
predictions.csv

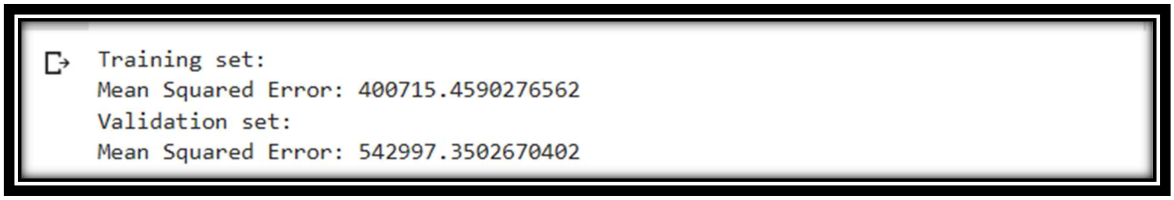


1 to 10 of 68720 entries

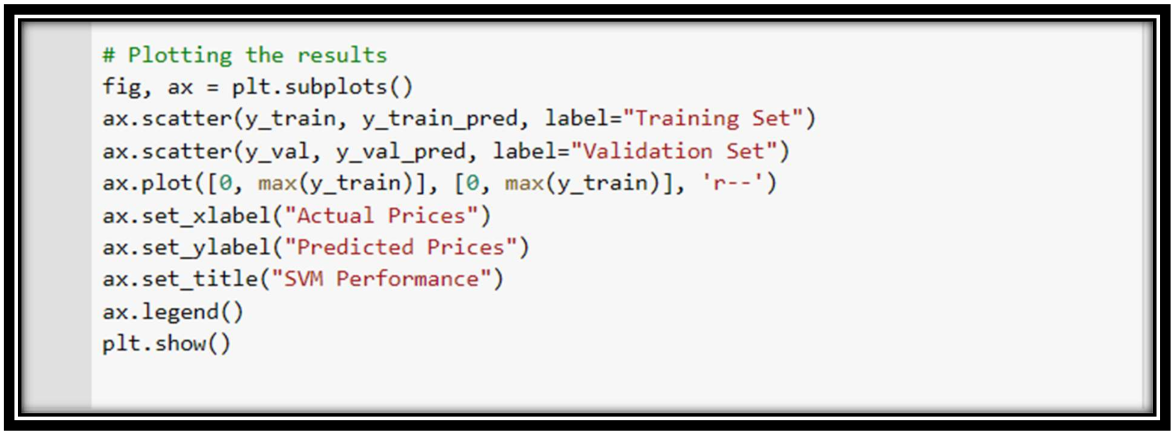
TARGET(PRICE_IN_LACS)
18.997391044240118
88.81793723869654
35.765643511979924
97.0644566569812
17.64381364789452
36.96952504833435
51.24301993535147
58.517512894941916
52.34062540674732
88.41752246908723



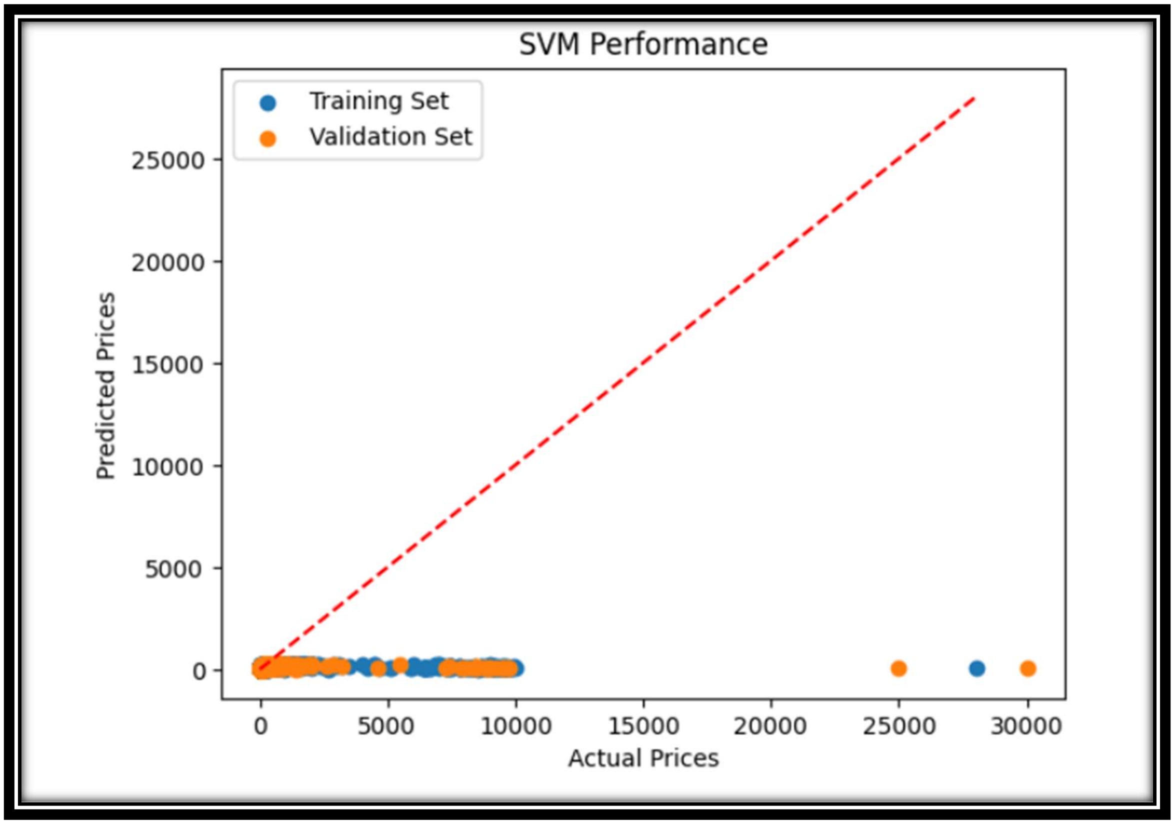
```
#SVM model
svm = SVR(kernel='rbf')
svm.fit(X_train, y_train)

#pred on the training and validation sets
y_train_pred = svm.predict(X_train)
y_val_pred = svm.predict(X_val)

train_mse = mean_squared_error(y_train, y_train_pred)
val_mse = mean_squared_error(y_val, y_val_pred)
print("Training set:")
print("Mean Squared Error:", train_mse)
print("Validation set:")
print("Mean Squared Error:", val_mse)
```

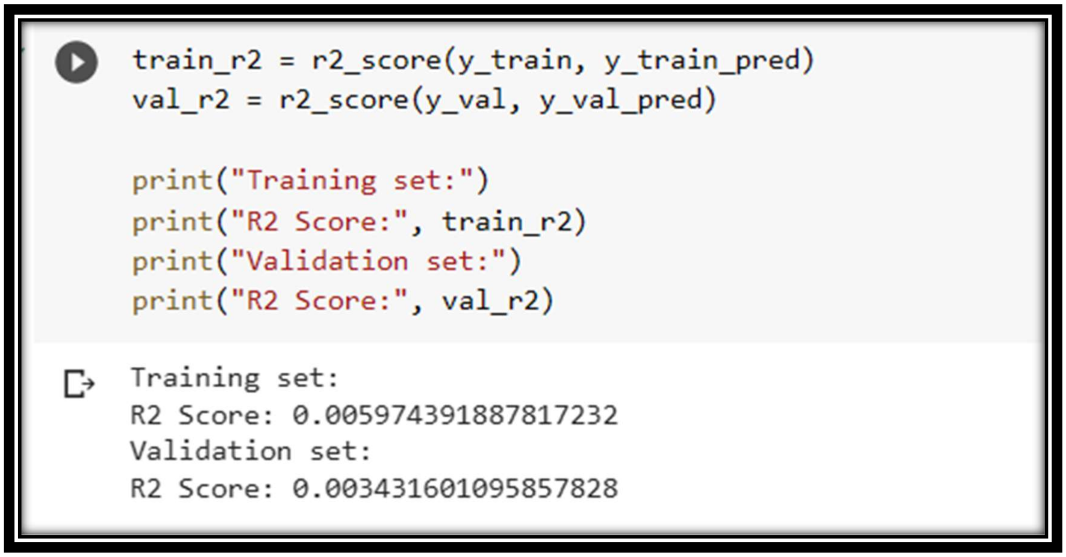


```
Training set:
Mean Squared Error: 400715.4590276562
Validation set:
Mean Squared Error: 542997.3502670402
```



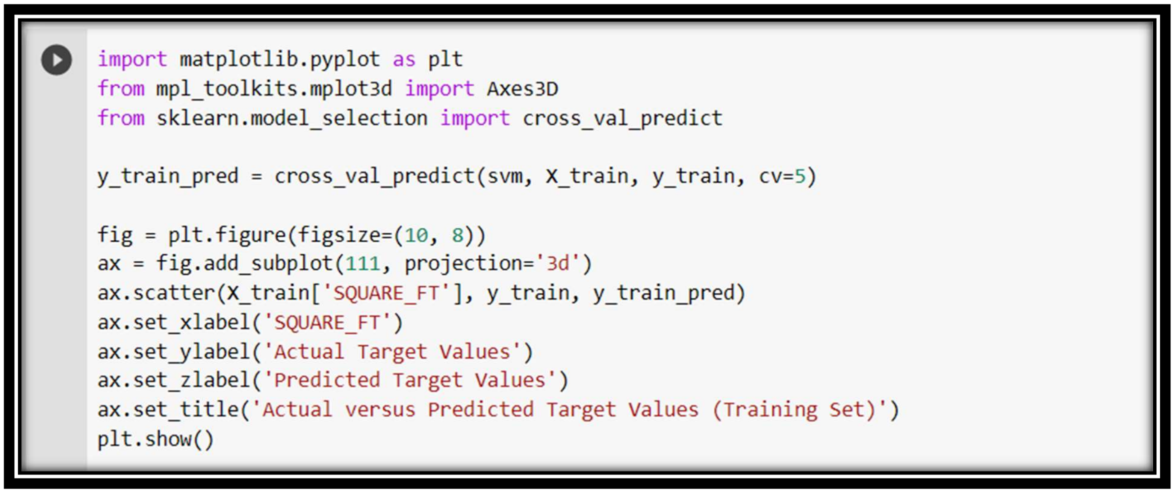
```
# Plotting the results
fig, ax = plt.subplots()
ax.scatter(y_train, y_train_pred, label="Training Set")
ax.scatter(y_val, y_val_pred, label="Validation Set")
ax.plot([0, max(y_train)], [0, max(y_train)], 'r--')
ax.set_xlabel("Actual Prices")
ax.set_ylabel("Predicted Prices")
ax.set_title("SVM Performance")
ax.legend()
plt.show()
```



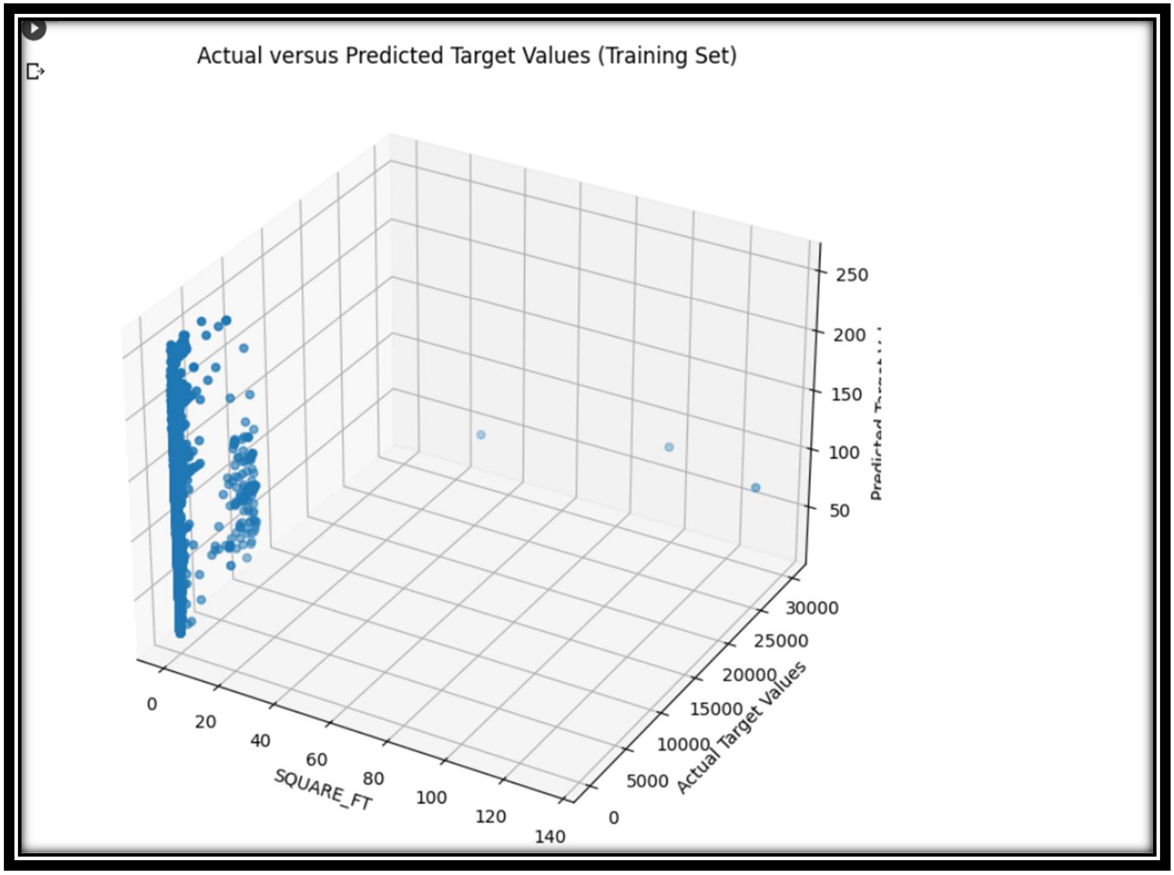


```
train_r2 = r2_score(y_train, y_train_pred)
val_r2 = r2_score(y_val, y_val_pred)

print("Training set:")
print("R2 Score:", train_r2)
print("Validation set:")
print("R2 Score:", val_r2)
```

```
Training set:
R2 Score: 0.005974391887817232
Validation set:
R2 Score: 0.003431601095857828
```



```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.model_selection import cross_val_predict

y_train_pred = cross_val_predict(svm, X_train, y_train, cv=5)

fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(X_train['SQUARE_FT'], y_train, y_train_pred)
ax.set_xlabel('SQUARE_FT')
ax.set_ylabel('Actual Target Values')
ax.set_zlabel('Predicted Target Values')
ax.set_title('Actual versus Predicted Target Values (Training Set)')
plt.show()
```


Actual versus Predicted Target Values (Training Set)
SQUARE_FT
Actual Target Values
Predicted Target