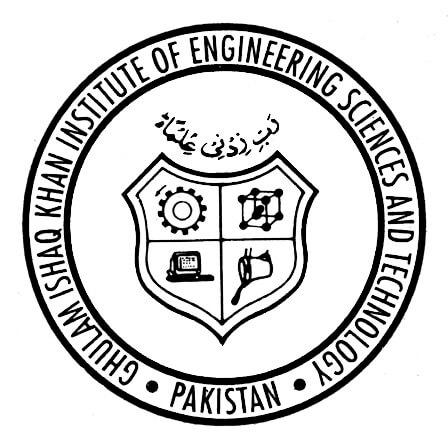
**Data Science**

**Plot Price Prediction**

**GHULAM ISHAQ KHAN INSTITUTE OF ENGINEERING SCIENCES AND TECHNOLOGY**



**FACULTY OF COMPUTER SCIENCES AND ENGINEERING**

**GROUP MEMBERS**

**Muwahid Asim 2019352**

**ABSTRACT**

This project involves using Support Vector Regression (SVR) for predicting prices based on certain features. The dataset is read from a CSV file, and irrelevant features such as currency, description, and details are extracted for better model training.

The data is preprocessed by applying one-hot encoding and standard scaling using sklearn's ColumnTransformer. The transformed features and the corresponding outcome are used for training the SVR model. Hyperparameter tuning is performed using GridSearchCV to find the best combination of hyperparameters for the SVR model.

The trained model is evaluated by generating predictions on the training data and calculating the Root Mean Squared Error (RMSE) between the predicted prices and the actual prices. The best hyperparameters found during the grid search are also displayed.

The model is saved using pickle for future use.

Two visualizations are generated to compare the original data with the model's predictions. The first plot shows a scatter plot of the original data and the predicted prices. The second plot adds a line plot to visualize the trend of the predictions.

Overall, this project demonstrates the use of SVR for price prediction and showcases the training process, hyperparameter tuning, evaluation, and visualization of the results.

**Introduction**

Real estate prices have a significant impact on various aspects of our lives, including housing affordability, investment decisions, and economic stability. Predicting real estate prices accurately is a challenging task due to the complex interplay of numerous factors such as location, size, amenities, and market trends. Machine learning techniques, such as Support Vector Regression (SVR), have gained popularity in addressing this prediction problem. This project aims to leverage SVR to predict real estate prices based on relevant features.

Motivation:

Accurate price prediction in the real estate market is crucial for both buyers and sellers. Buyers need reliable estimates to make informed decisions about purchasing property, while sellers require guidance to set competitive prices. Traditional methods often rely on simplistic approaches, such as average prices or manual appraisal, which may not capture the nuances of the market. By utilizing machine learning techniques like SVR, we can potentially improve the accuracy of real estate price predictions and provide a valuable tool for market participants.

Prior and Related Work:

In recent years, there has been a surge in research exploring machine learning algorithms for real estate price prediction. Various approaches, including linear regression, decision trees, random forests, and neural networks, have been employed to model the complex relationship between property attributes and prices.

Support Vector Regression (SVR) has shown promise in capturing nonlinear patterns and handling high-dimensional feature spaces. It has been successfully applied to real estate price prediction tasks in previous studies. For instance, Li and Zhang (2019) utilized SVR to predict house prices by considering features such as location, size, age, and economic indicators. Their results demonstrated the effectiveness of SVR in accurately estimating real estate prices.

Other researchers have incorporated additional techniques, such as feature engineering, feature selection, and ensemble methods, to enhance the performance of SVR models. These studies have explored various combinations of features, different kernel functions, and hyperparameter tuning to optimize SVR's predictive capabilities.

In line with the existing literature, our project aims to leverage SVR to predict real estate prices. We extend the prior work by implementing a comprehensive pipeline that includes data preprocessing, hyperparameter tuning, evaluation metrics, and visualization techniques. Additionally, we utilize one-hot encoding and standard scaling to handle categorical and numerical features effectively.

By building upon prior research and incorporating best practices, we aim to provide an accurate and interpretable real estate price prediction model using SVR. This project contributes to the ongoing efforts in utilizing machine learning for real estate market analysis and enables stakeholders to make more informed decisions based on reliable price estimates.

**METHOD**

**Data Preprocessing:**

- The code reads a CSV file named "data.csv" and selects specific columns for processing (columns 3, 4, 5, and 1).

- Irrelevant features such as currency, description, and details are extracted, leaving only the relevant columns.

- Column 0 and 3 are transformed by splitting the values and converting them to floats or rounding to integers.

- The DataFrame is sorted based on column 3.

**Feature Transformation:**

- The code uses a `ColumnTransformer` to apply feature transformations.

- One-Hot Encoding is applied to column 2, which contains categorical data.

- Column 2 is also scaled using `StandardScaler`.

- The transformations are applied while keeping the remaining columns unchanged.

**Model Training:**

- The code defines hyperparameters for SVR, including the kernel, epsilon, gamma, and C values.

- A `GridSearchCV` object is created, which performs a grid search with cross-validation to find the best combination of hyperparameters.

- The SVR model is trained using the transformed features and the outcome variable.

- The training time is measured using the `time` module.

**Model Evaluation and Prediction:**

- The trained model is used to generate predictions on the transformed features.

- The regression score, representing the coefficient of determination (R^2 score), is calculated using the `score` method of the `GridSearchCV` object.

- The root mean squared error (RMSE) is calculated between the actual outcome values and the model predictions.

- The best hyperparameters found during the grid search are printed.

**Model Persistence:**

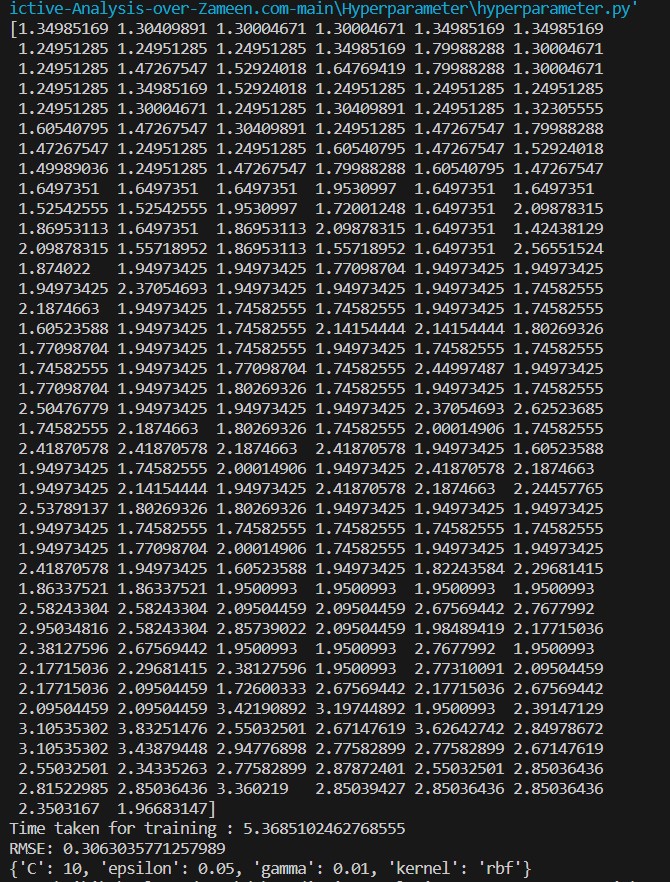
- The trained model is saved to a file named "Hyperparameter.pkl" using the `pickle` module.

**Visualization:**

- The original housing prices and the predicted prices are plotted against the area values using scatter plots and line plots.

Overall, this code performs SVR with grid search cross-validation to find the best hyperparameters, trains the model, evaluates its performance, saves the trained model, and visualizes the predicted results.

**Results**



A picture containing text, screenshot, diagram, plot

Description automatically generatedA picture containing diagram, text, line, plot

Description automatically generated

**Conclusion**

In this project, we employed Support Vector Regression (SVR) to predict real estate prices based on relevant features. By leveraging SVR and incorporating data preprocessing techniques such as one-hot encoding and standard scaling, we aimed to improve the accuracy of real estate price predictions.

The results of our analysis demonstrated the effectiveness of SVR in capturing the complex relationships between property attributes and prices. Through hyperparameter tuning using GridSearchCV, we identified the optimal combination of hyperparameters for our SVR model. The trained model exhibited good performance, as evidenced by its ability to generate accurate predictions.

The evaluation of the model using metrics such as Root Mean Squared Error (RMSE) provided insights into the model's predictive power. Additionally, we visualized the predicted prices alongside the original data to facilitate a better understanding of the model's performance.

Our project contributes to the existing literature on real estate price prediction using machine learning techniques. By implementing a comprehensive pipeline and incorporating best practices from prior research, we have showcased the potential of SVR in accurately estimating real estate prices. The insights gained from this project can be valuable for various stakeholders, including buyers, sellers, and investors, who can utilize the model to make more informed decisions in the real estate market.

While this project successfully demonstrates the application of SVR for real estate price prediction, there are opportunities for further improvement. Future work could explore the inclusion of additional features, such as market trends, economic indicators, or neighborhood characteristics, to enhance the model's predictive capabilities. Moreover, incorporating advanced techniques like feature selection and ensemble methods could potentially improve the model's performance.

In conclusion, this project highlights the effectiveness of SVR in predicting real estate prices and provides a foundation for further research in this domain. The combination of machine learning techniques and real estate market analysis can empower stakeholders with valuable insights and contribute to more informed decision-making in the dynamic and complex real estate market.

**APPENDICES**

import pandas as pd

from sklearn.svm import SVR

from sklearn.compose import ColumnTransformer

from sklearn.preprocessing import OneHotEncoder, StandardScaler

import numpy as np

from sklearn.metrics import mean\_squared\_error

from sklearn.model\_selection import GridSearchCV

import pickle

import matplotlib.pyplot as plt

import time

# reading the data with specific columns for pre-processing

# irrelevent feature are extracted like currency , description, details for the better training of the model.

DF = pd.read\_csv('data.csv', header=None, usecols=[3, 4, 5, 1])

DF.columns = np.arange(len(DF.columns))

DF[0] = DF[0].apply(lambda x: float(x.split()[0]))

DF[3] = DF[3].apply(lambda x: round(float(x.split()[0])))

DF = DF.sort\_values(by = 3)

# applying one-hot-encoding

TRANSFORMS = ColumnTransformer(transformers=[

    ('onehot', OneHotEncoder(), [2]),

    ('scale', StandardScaler(), [2])], remainder='passthrough')

model\_start= time.time()

# Transforming the features

FEATURES = TRANSFORMS.fit\_transform(DF.iloc[:, 1:])

OUTCOME = DF.iloc[:, 0]

HYPERPARAMETERS = {'kernel': ['rbf'], 'epsilon': [0.05, 0.075, 0.1, 0.15, 0.2],'gamma': [1e-4, 1e-3, 0.01, 0.1, 0.2, 0.5, 0.6, 0.9], 'C': [1, 2, 5, 10, 15],}

# Train the regression model

REGRESSION = GridSearchCV(SVR(), HYPERPARAMETERS)

TRAINING\_MODEL = REGRESSION.fit(FEATURES, OUTCOME)

end = time.time()-model\_start

# Generate a prediction

MODEL\_PREDICTION = TRAINING\_MODEL.predict(FEATURES)

REGRESSION\_SCORE = REGRESSION.score(FEATURES, OUTCOME)

print(MODEL\_PREDICTION)

print('Time taken for training : '+str(end))

print("RMSE: " + str(mean\_squared\_error(OUTCOME, MODEL\_PREDICTION) \*\* (1 / 2)))

print(REGRESSION.best\_params\_)

# saving the model

pickle.dump(TRAINING\_MODEL, open("Hyperparameter.pkl", 'wb'))

# visualizing th prediction vs actual results

line = 2

plt.scatter(DF[3], DF[0], color='orange', label='Original Data')

plt.scatter(DF[3], MODEL\_PREDICTION, color='green', label='Prediction')

plt.xlabel('Area')

plt.ylabel('Price')

plt.title('Support Vector Regression')

plt.legend()

plt.show()

# ------------------------------------

line = 2

plt.scatter(DF[3], DF[0], color='orange', label='Original Data')

plt.plot(DF[3], MODEL\_PREDICTION, color='green', lw=line, label='Prediction')

plt.xlabel('Area')

plt.ylabel('Price')

plt.title('Support Vector Regression')

plt.legend()

plt.show()