CSE 281:
Introduction to Artificial Intelligence

Project Report

DECEMBER 27

Authored by:

Abdelrahman Mohamed Zayed	**23P0094**
Hassan Hazem Abdelhady	**23P0415**
Ahmad ElSayed Abdelhafez	**23P0018**
Jana Mohamed Wael	**23P0200**
Mariam Mohamed Shemies	**23P0178**
Mostafa Amr Mostafa	**23P0206**

Contents

Abstract

Accurate prediction of item prices is a vital aspect of market analysis, enabling businesses to optimize pricing strategies, forecast revenues, and respond to market dynamics. This project focuses on developing predictive models using multiple regression techniques. A provided dataset containing numerical and categorical features serves as the basis for model development. The project involves data preprocessing, exploratory data analysis, and the application of four distinct regression models: Support Vector Regression (SVR), XGBRegressor, Random Forest Regressor, and Linear Regression. The aim is to achieve high predictive accuracy by minimizing errors and adhering to rigorous methodological guidelines. This report provides a comprehensive overview of the dataset, methodologies, model performances, and insights gained.

Introduction

Predicting item prices is a critical task in various industries, from retail to real estate, where accurate price predictions directly impact strategic decision-making and competitive advantage. The task involves analyzing historical data to uncover patterns and relationships that influence pricing, such as product characteristics, regional factors, and market conditions. By leveraging machine learning techniques, businesses can improve efficiency, enhance customer satisfaction, and stay competitive in dynamic markets.

This project, conducted as part of the CSE281 course, aims to equip us with practical skills in predictive modeling and data science. The dataset provided comprises a mix of numerical and categorical features, offering a realistic challenge that reflects real-world scenarios. The project requires :
- Explore and preprocess the dataset.
- Implement multiple regression models to predict item prices.
- Evaluate model performance based on error metrics.
- Document their findings and methodologies comprehensively.

By combining technical rigor with a focus on interpretability, this project not only builds foundational knowledge in regression analysis but also fosters the ability to draw actionable insights from data. Through the application of Support Vector Regression, XGBRegressor, Random Forest Regressor, and Linear Regression.

Project Objective

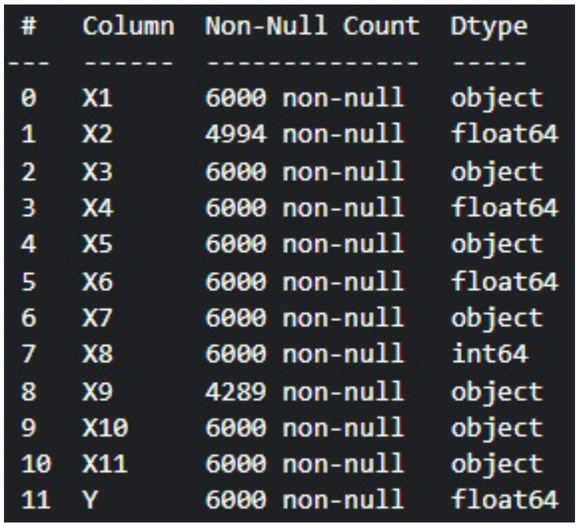
The primary goal of this project is to develop multiple regression models to predict item prices based on a provided dataset. By analyzing the dataset, applying preprocessing techniques, and leveraging regression models, the objective is to minimize prediction errors and create a robust solution. The project is designed to foster skills in data analysis, model development, and evaluation, as well as adherence to methodological rigor.

Steps and Implementation

1. Data Analysis

1.1 Looking for Null values.

The data is analyzed in search for NULL values, which we can see by using df.info().

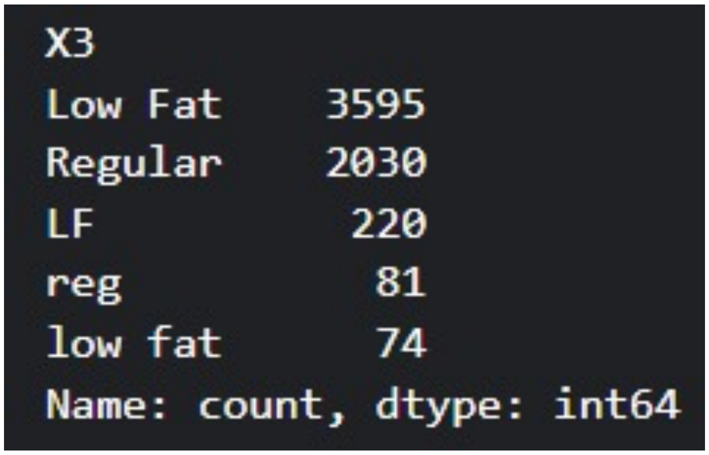


```
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   X1      6000 non-null   object 
 1   X2      4994 non-null   float64
 2   X3      6000 non-null   object 
 3   X4      6000 non-null   float64
 4   X5      6000 non-null   object 
 5   X6      6000 non-null   float64
 6   X7      6000 non-null   object 
 7   X8      6000 non-null   int64  
 8   X9      4289 non-null   object 
 9   X10     6000 non-null   object 
 10  X11     6000 non-null   object 
 11  Y       6000 non-null   float64
```

We can verify that there are missing values in both X2 and X9.

1.2 Looking for mistakes

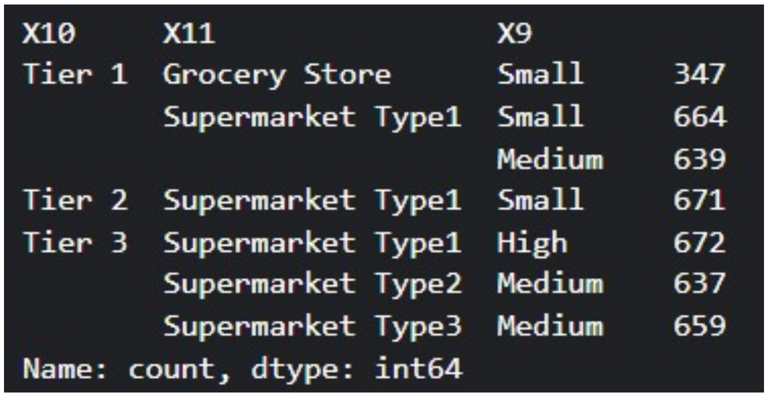
The X3 column, which stands for fat content of a product, has a lot of misspellings that need to be corrected to ensure consistency.



```
X3
Low Fat    3595
Regular    2030
LF          220
reg          81
low fat      74
Name: count, dtype: int64
```

1.3 Analyzing relations between columns

The X9, X10 and X11 columns, that represent size of store, location tier and type of store respectively are heavily related to each other.



```
X10     X11                X9
Tier 1  Grocery Store      Small     347
        Supermarket Type1  Small     664
                           Medium    639
Tier 2  Supermarket Type1  Small     671
Tier 3  Supermarket Type1  High      672
        Supermarket Type2  Medium    637
        Supermarket Type3  Medium    659
Name: count, dtype: int64
```

We can see that tier 1 has small and medium attribute shops, that are either a Grocery store or type 1 Supermarket. Tier 2 has small attribute shops, that are type 1 Supermarkets. Tier 3 has 2-medium attribute shops (that are of type 2 and 3) and 1 high attribute shop that is of type 1.

2. Data Preprocessing

2.1 Converting Year to Age

The X8 column, which represents a year, was transformed into an age column using the formula: Age=2024−X8 , this was done to make the range of the values closer to the rest of the data.

2.2 Transforming Product Type (X1)

The X1 column was mapped to product categories based on the first two letters:

FD → "Food"

DR → "Drink"

NC → "Non-Consumable"

2.3 Standardizing X3 Column

To ensure consistency, the values in the X3 column were updated as follows:

"low fat" → "Low Fat"

"LF" → "Low Fat"

"reg" → "Regular"

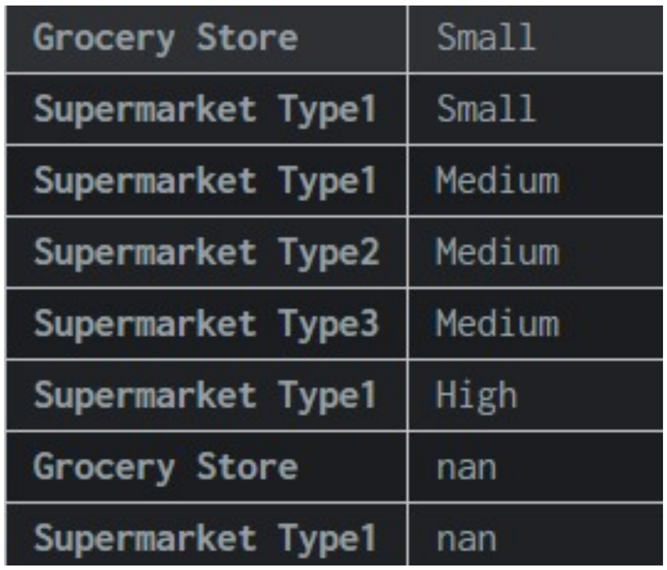
2.4 Imputing Missing Values for X2

We impute the missing values in X2 with the mean because it is a continuous numerical variable, and the mean provides a simple and effective way to represent the central tendency of the data. This method ensures that the imputed values align with the overall distribution of X2, minimizing the distortion of statistical properties like the average and variance. Using the mean is particularly appropriate when the data is symmetrically distributed and free from significant outliers, as it maintains consistency without introducing bias. Additionally, imputing with the mean is computationally efficient and helps retain the dataset's usability for further analysis or modeling.

We used KNN Imputer to impute the missing values in X2, KNN imputer is useful for handling missing data by imputing missing values based on the nearest neighbors. It works by finding similar instances (neighbors) in the dataset and using their feature values to predict the missing values, making it effective for datasets with non-linear relationships. This method is especially helpful when the missing data is not missing at random and is likely correlated with other features, allowing for more accurate imputations compared to simple methods like mean or median imputation.

2.5 Imputing Missing Values for X9

Missing values in the X9 column were filled using the mode of X9, grouped by the X11 column. This ensured that the imputed values aligned with the grouping characteristics of X11.



Grocery Store	Small
Supermarket Type1	Small
Supermarket Type1	Medium
Supermarket Type2	Medium
Supermarket Type3	Medium
Supermarket Type1	High
Grocery Store	nan
Supermarket Type1	nan

2.6 Encoding Categorical Columns

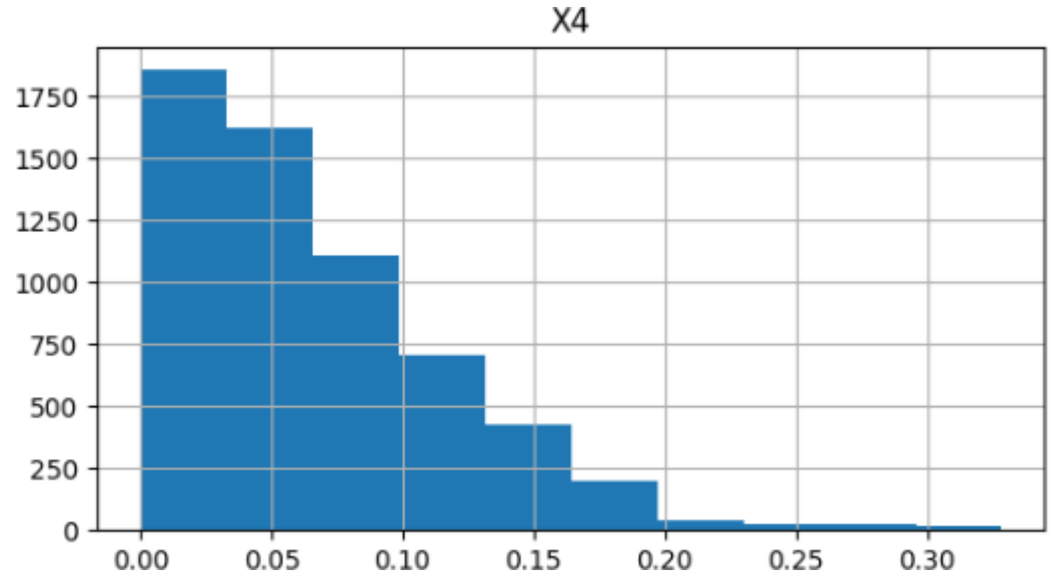
Ordinal Columns: Categorical columns with inherent order were encoded using Label Encoding.

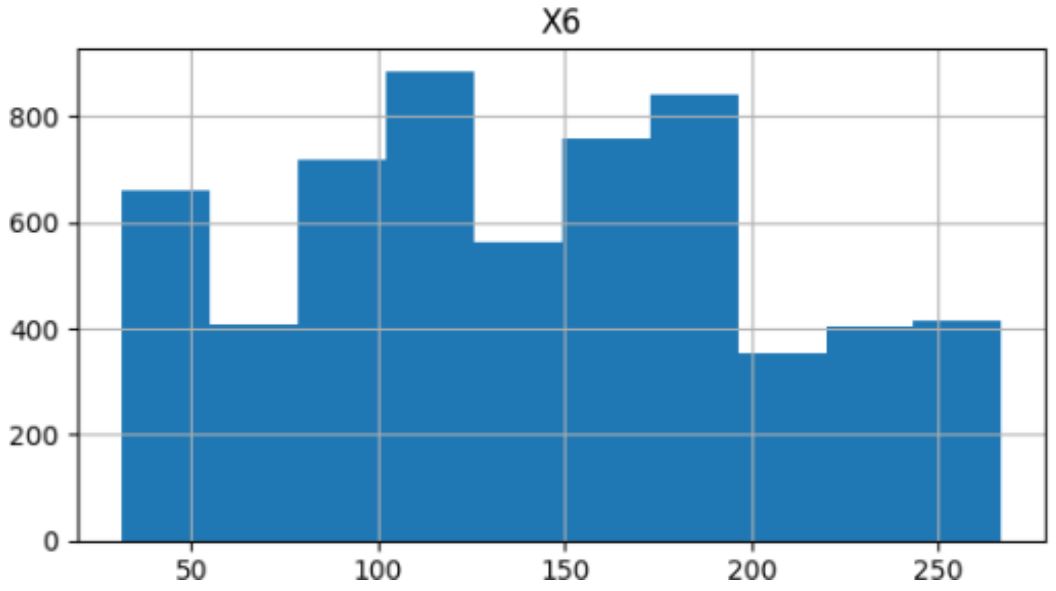
Nominal Columns: Columns without order were encoded using One-Hot Encoding.

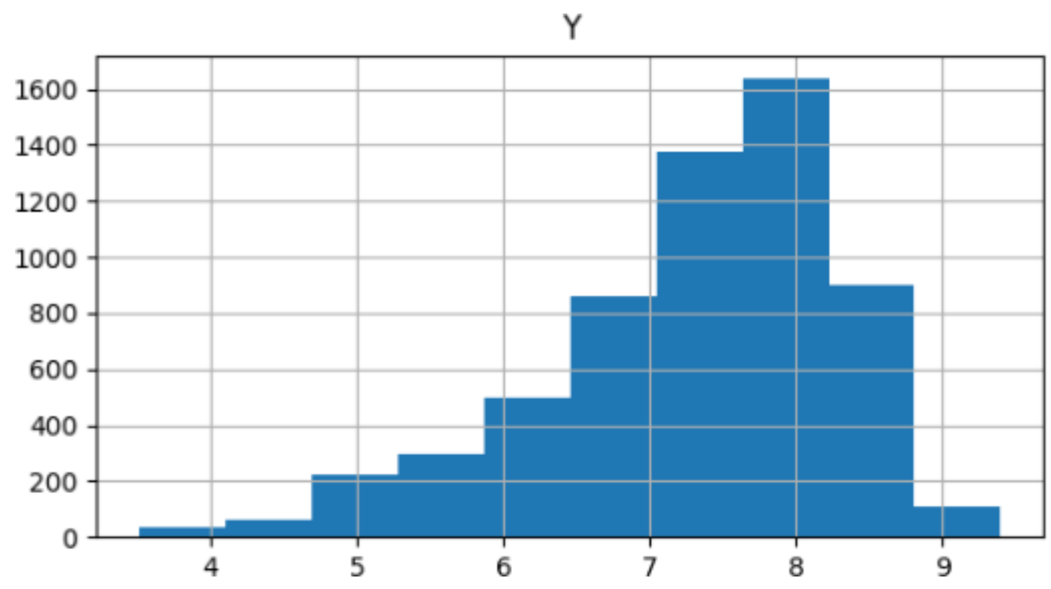
Additionally, ordinal columns were scaled to reflect their ordered nature and improve model performance.

2.7 Scaling Numerical Columns

To normalize numerical columns, MinMaxScaler was applied, as the data does not follow a Gaussian distribution, as shown in these histograms:



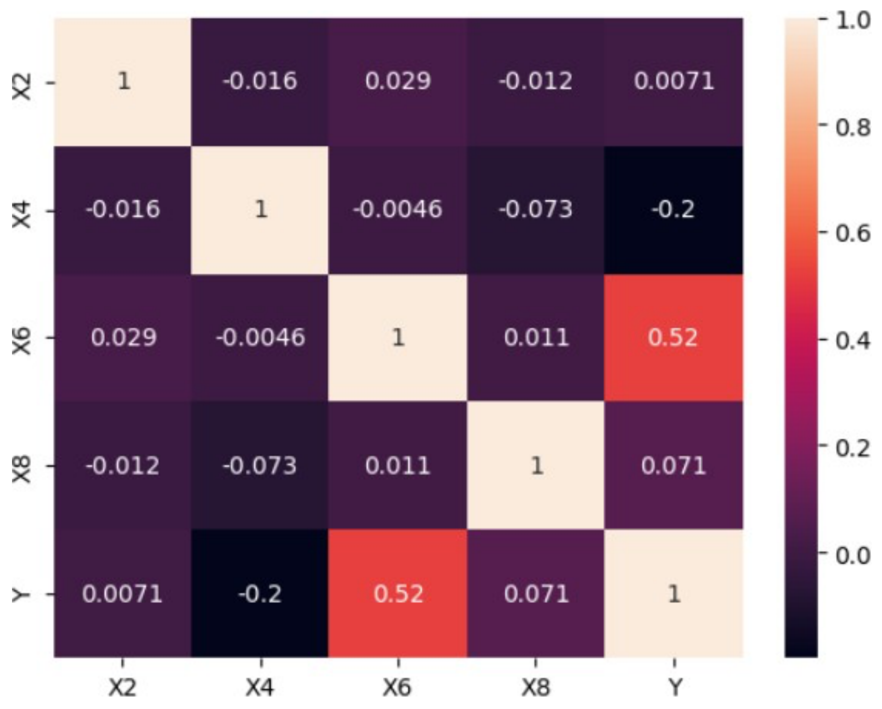




3. Feature Engineering

3.1 Correlation Analysis

A heatmap was generated to examine the correlations between all features and the target column (Y). This analysis helped identify features with significant relationships to the target variable.



	X2	X4	X6	X8	Y
X2	1	-0.016	0.029	-0.012	0.0071
X4	-0.016	1	-0.0046	-0.073	-0.2
X6	0.029	-0.0046	1	0.011	0.52
X8	-0.012	-0.073	0.011	1	0.071
Y	0.0071	-0.2	0.52	0.071	1

3.2 Creating Non-Linear Features

New features were generated by multiplying selected columns to capture potential non-linear interactions.

3.3 Feature Selection

Features were shortlisted based on their correlation strength with the target variable to ensure only the most relevant predictors were included in the model.

Model Selection and Hyperparameter Optimization

During our project, we dedicated significant effort to optimizing the parameters of various regression models using grid search. This process involved systematically evaluating different combinations of hyperparameters to identify the configurations that delivered the best performance. Below are the details of the models and the hyperparameters we identified:

1. Support Vector Regression (SVR)

- **Model 1: SVR(kernel='rbf', C=1000, epsilon=0.001)**
    - **Achieved a performance metric of 0.368.**
    - **This configuration uses a high penalty parameter (C) and a small epsilon value to ensure a precise fit for the data.**
- **Model 2: SVR(kernel='rbf', C=200, epsilon=0.0000001, gamma='scale')**
    - **Achieved a slightly less performance metric of 0.369 but was a better overall performance metric on the full dataset.**
    - **This configuration uses a lower C value and an even smaller epsilon, combined with an automatic (scale) calculation for the kernel coefficient (gamma).**

The incremental improvement between the two configurations highlights the impact of fine-tuning SVR hyperparameters.

2. Random Forest Regressor (RF)

Using grid search, we selected the following configuration:

- **RandomForestRegressor(n_estimators=100, random_state=321)**

This model combines the predictions of 100 decision trees, ensuring robust performance and reduced overfitting. The random_state parameter guarantees reproducibility of results.

3. XGB Regressor (XGB)

We optimized the XGBoost model using the following hyperparameters:

- **XGBRegressor(n_estimators=100, learning_rate=0.1, max_depth=3, objective='reg:absoluteerror')**

The learning rate (0.1) ensures steady convergence, while the max_depth parameter (3) limits the complexity of the model, balancing bias and variance. The objective function (reg:absoluteerror) minimizes the absolute error to account for outliers effectively.

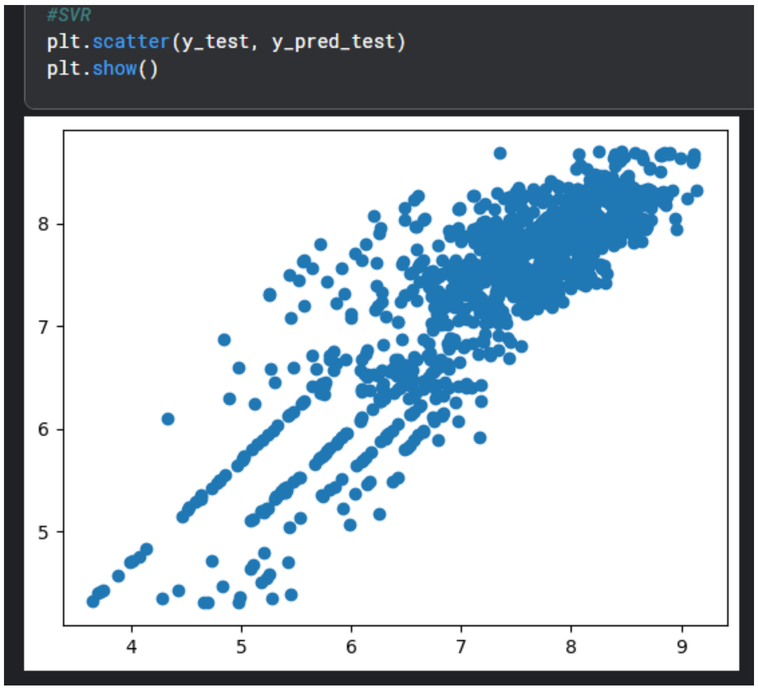
4. Linear Regression (Baseline)

Linear regression was used as a baseline model for comparison. Since it is parameter-free, no additional hyperparameter tuning was necessary.

Model Training and Prediction

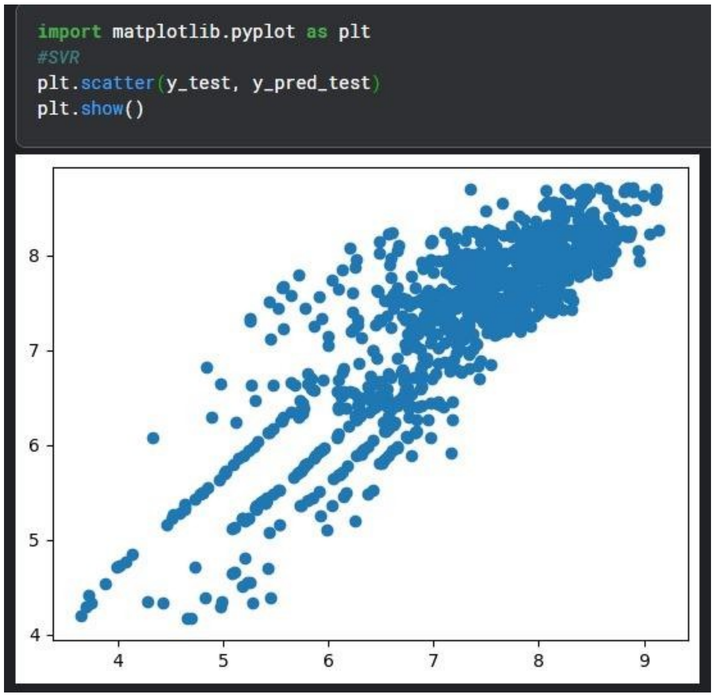
The following regression models were trained and evaluated on the dataset:

- **Support Vector Regression (SVR): A model that uses kernel functions to capture complex relationships.**
- **These are the submissions we chose for the SVR:**



```
#SVR
plt.scatter(y_test, y_pred_test)
plt.show()
```

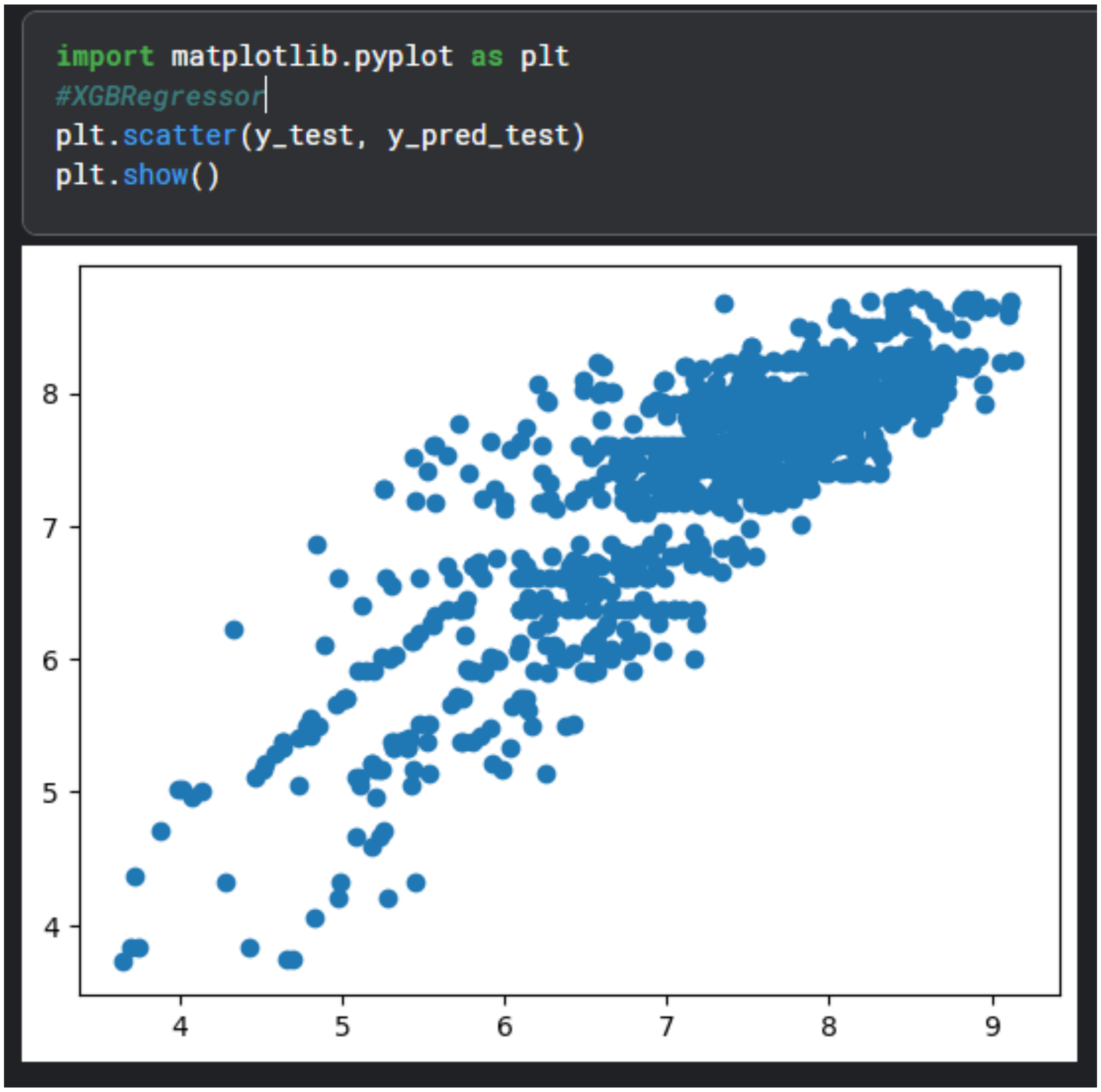
Figure 1: with score of 0.3C3



```
import matplotlib.pyplot as plt
#SVR
plt.scatter(y_test, y_pred_test)
plt.show()
```

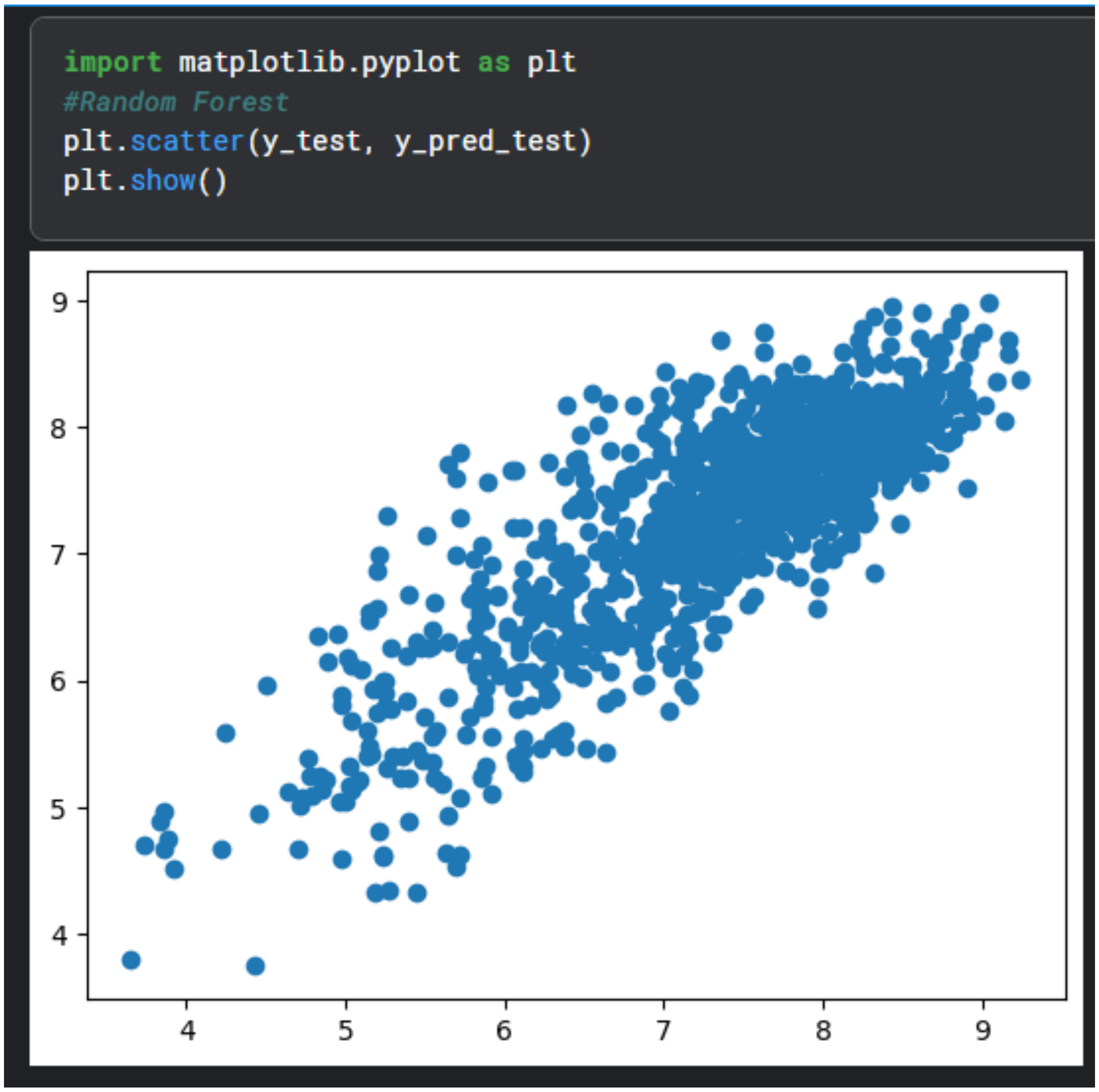
Figure 2: with score of 0.3C8

- **XGBRegressor (Extreme Gradient Boosting): A powerful gradient-boosting framework for structured data.**



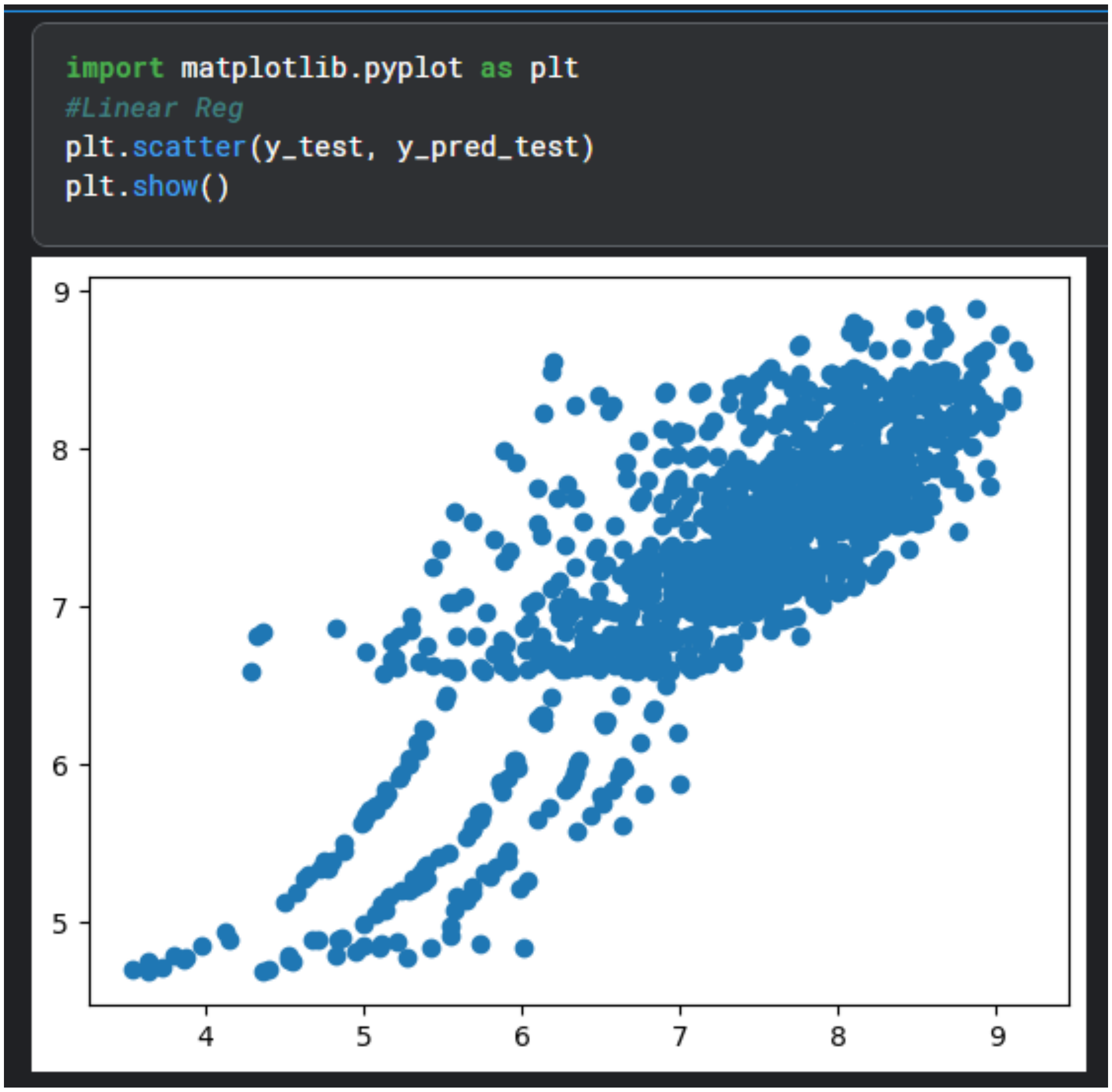
```
import matplotlib.pyplot as plt
#XGBRegressor
plt.scatter(y_test, y_pred_test)
plt.show()
```

- **Random Forest Regressor: An ensemble model that combines multiple decision trees for robust predictions.**



```
import matplotlib.pyplot as plt
#Random Forest
plt.scatter(y_test, y_pred_test)
plt.show()
```

- **Linear Regression: A baseline model for benchmarking performance.**



```
import matplotlib.pyplot as plt
#Linear Reg
plt.scatter(y_test, y_pred_test)
plt.show()
```

Each model was trained on the preprocessed dataset, and predictions were generated for the target variable (Y). Performance metrics such as Mean Absolute Error (MAE) were calculated to compare the models and select the best-performing approach.

Mean Absolute Error was calculated using a series of train test splits and K-fold cross validations to ensure the validity of the MAE.

Conclusion

This project provided an opportunity to apply various regression models, analyze their performance, and gain practical experience in predictive modeling. By following the outlined requirements and utilizing diverse techniques, we demonstrated our ability to preprocess data, implement and evaluate models, and present findings comprehensively.