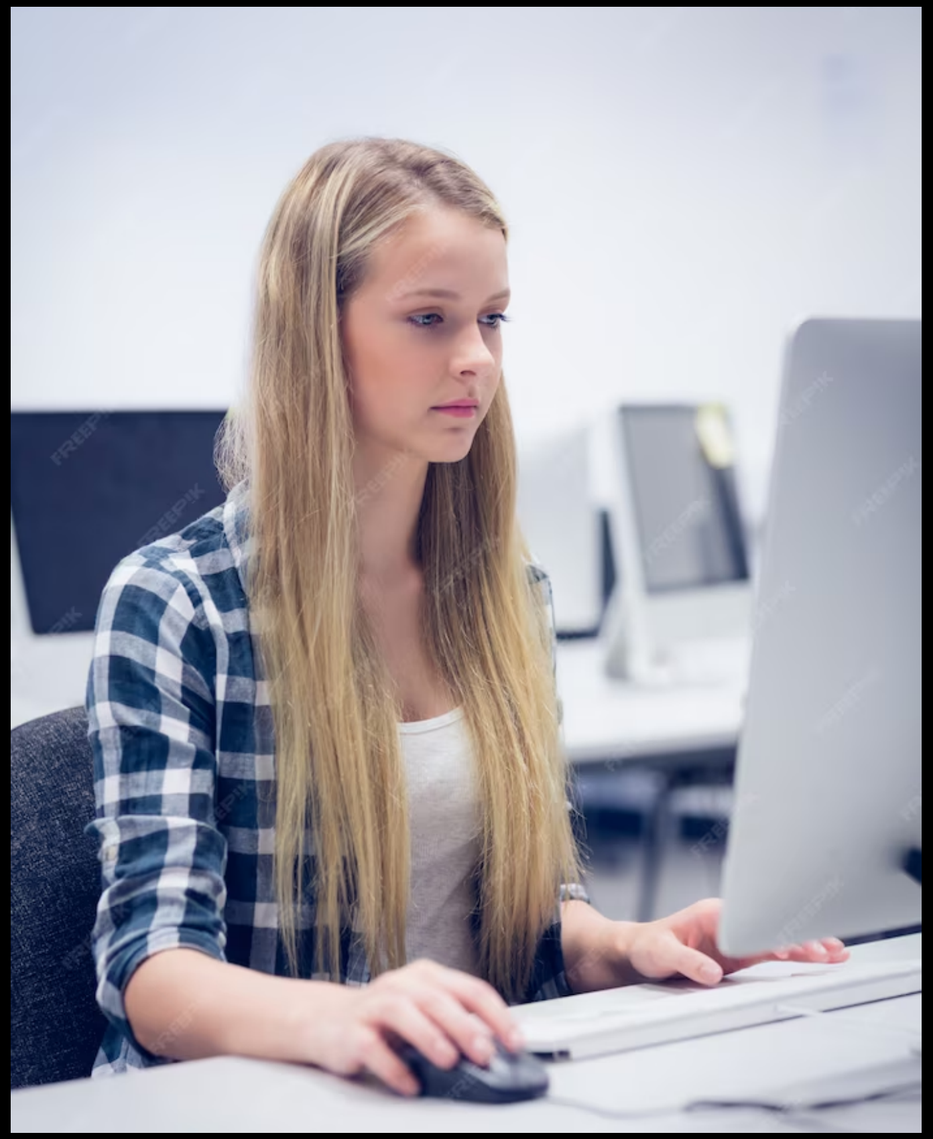
Enhancing Predictive Modeling Accuracy through Regression Analysis: A Comprehensive Evaluation



Introduction

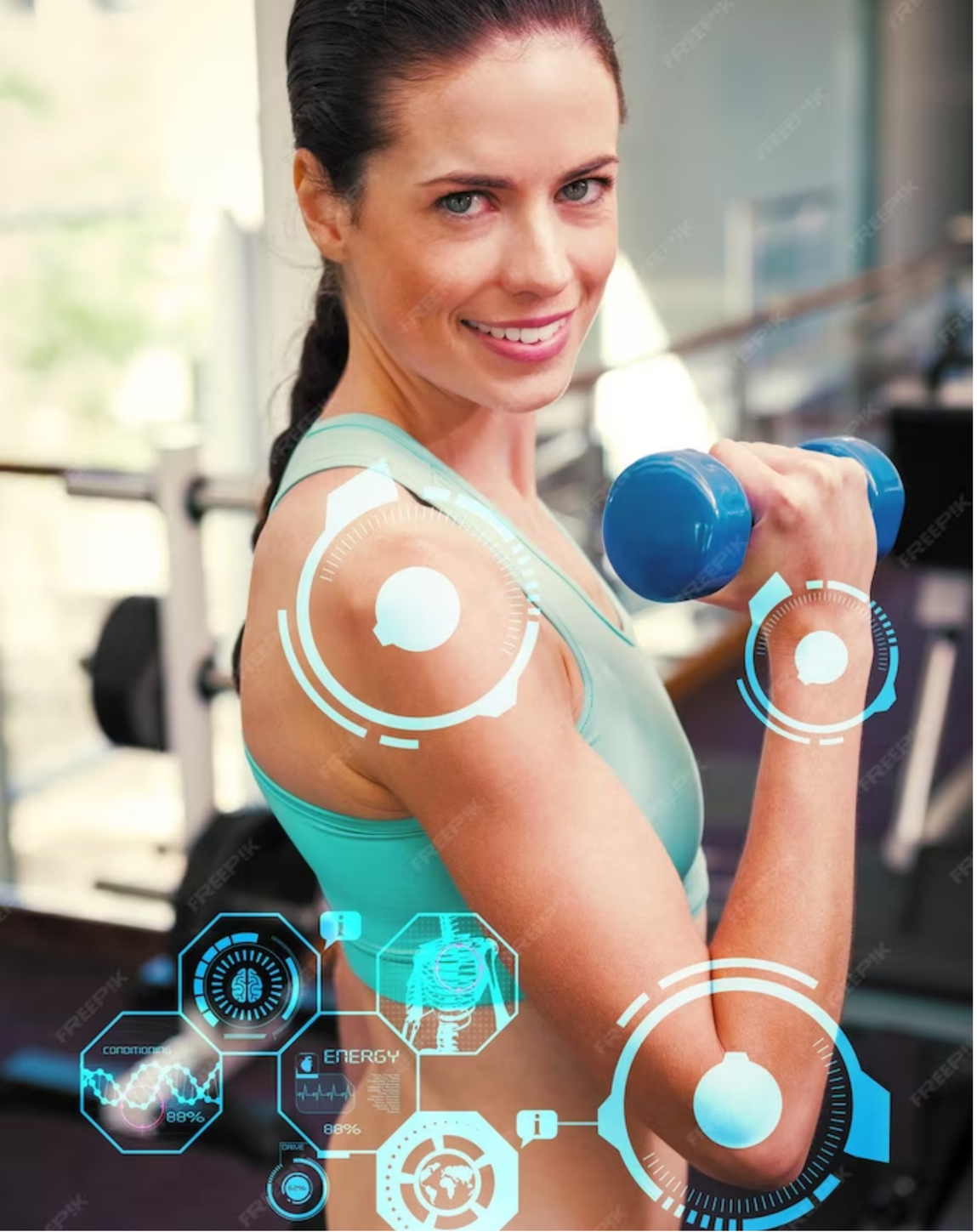
This presentation will discuss the importance of regression analysis in enhancing predictive modeling accuracy. We will cover the basics of regression analysis, its applications, and how it can be used to improve predictive modeling. By the end of this presentation, you will have a comprehensive understanding of regression analysis and its impact on predictive modeling accuracy.

What is Regression Analysis?

Regression analysis is a statistical method used to establish a relationship between a dependent variable and one or more independent variables. It is commonly used in predictive modeling to identify the relationship between the target variable and the predictor variables. Regression analysis helps in understanding the impact of predictor variables on the target variable.

Applications of Regression Analysis

Regression analysis has a wide range of applications in various fields, including finance, healthcare, marketing, and social sciences. It is used to predict stock prices, diagnose diseases, forecast sales, and analyze customer behavior. Regression analysis is a powerful tool that helps in making informed decisions based on data.

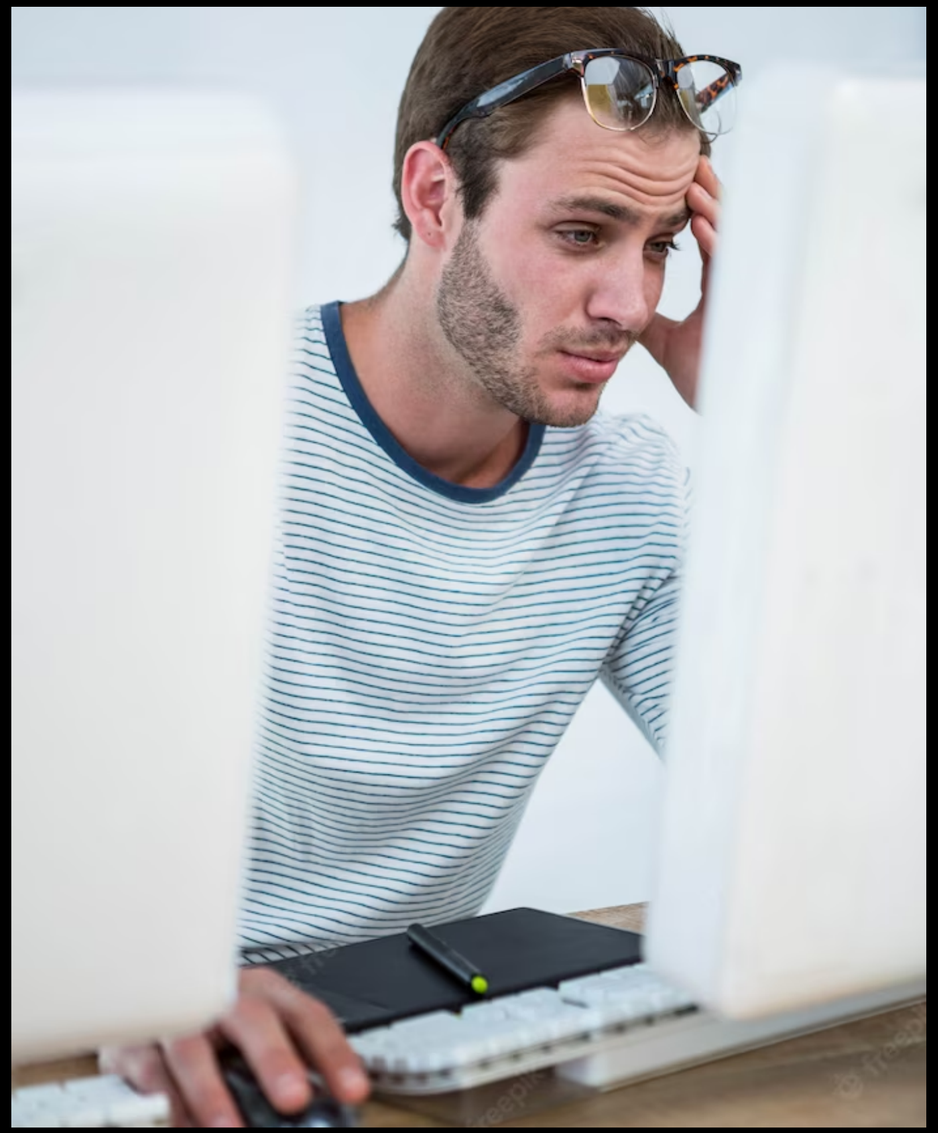


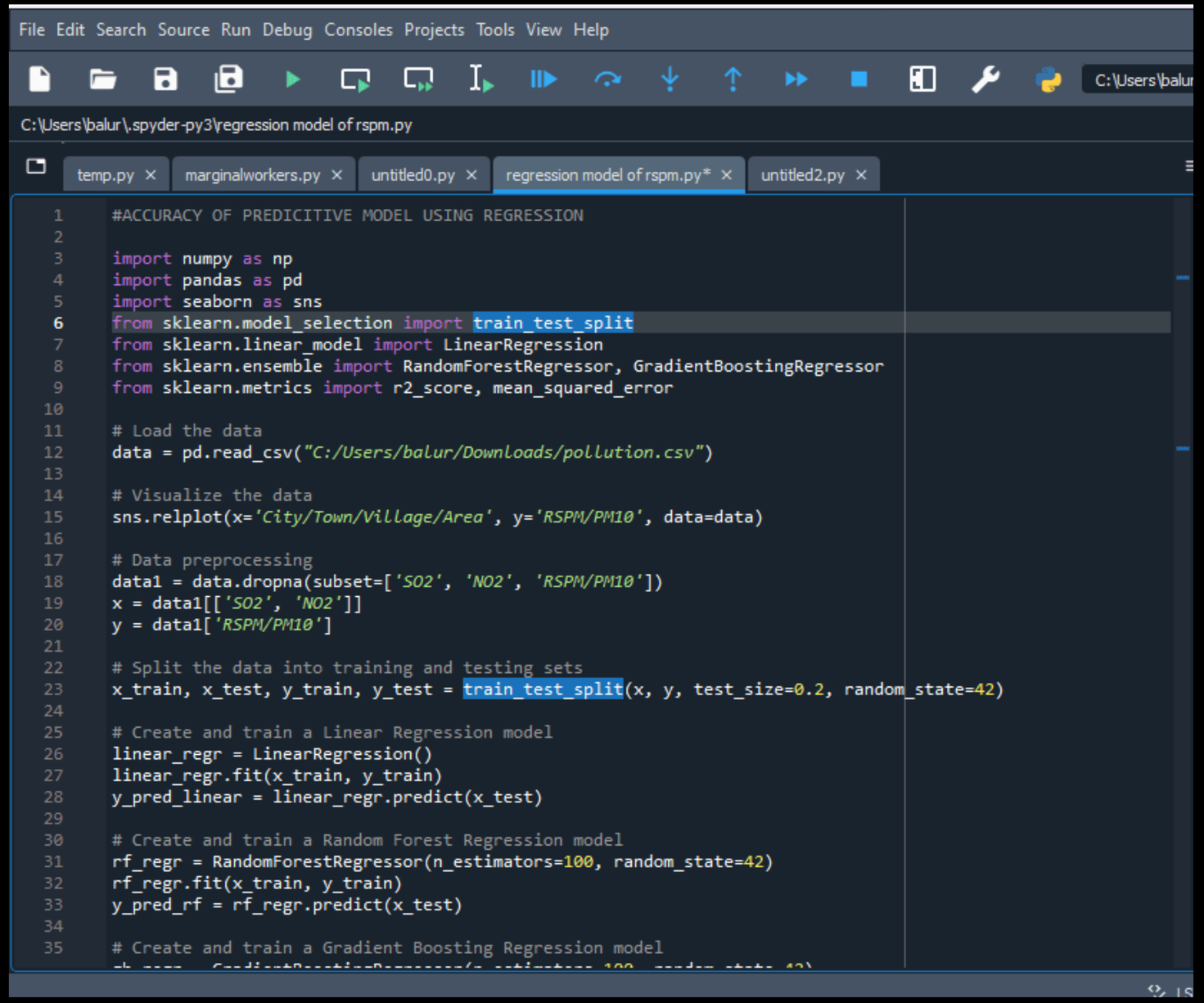
Enhancing Predictive Modeling Accuracy

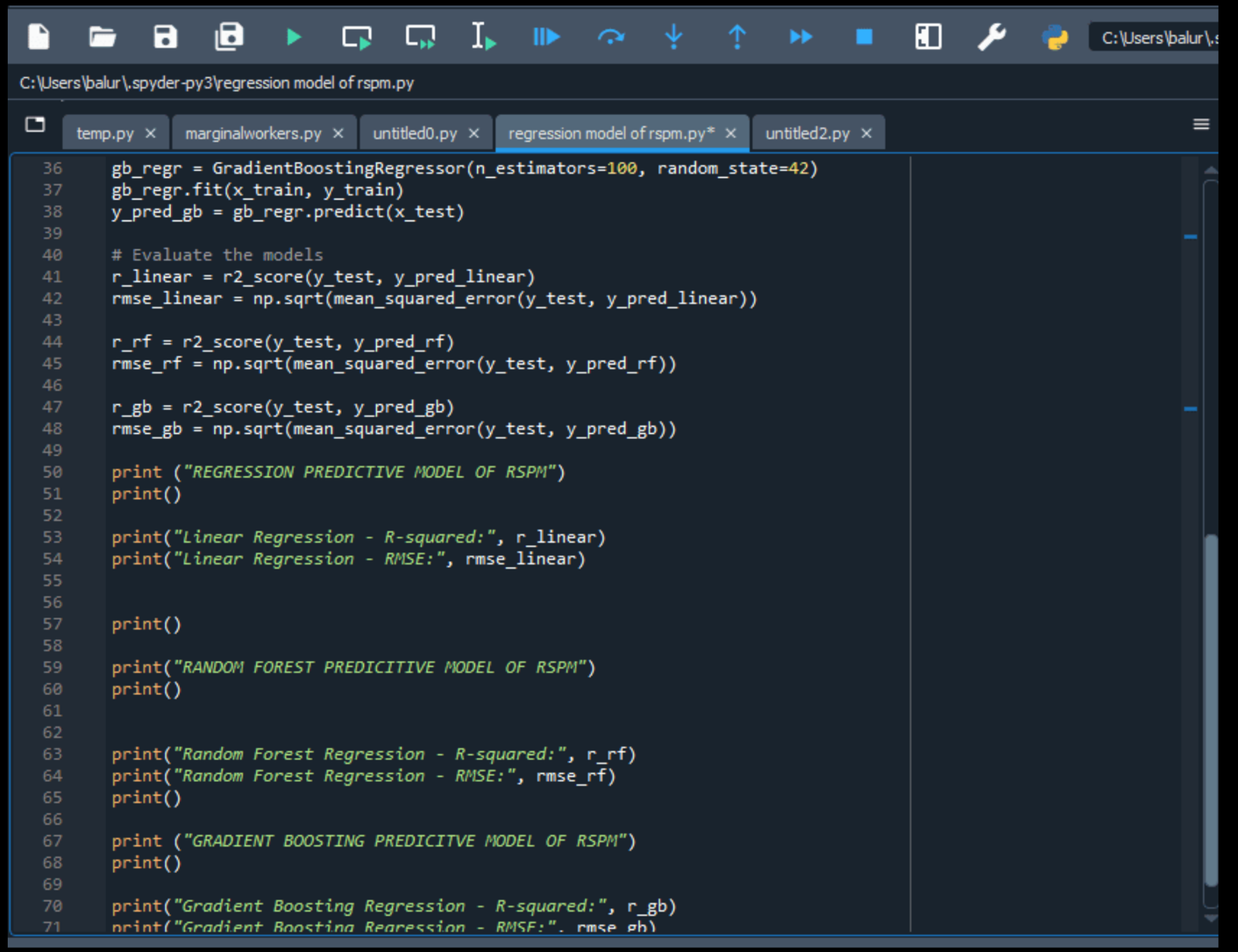
Regression analysis can be used to enhance predictive modeling accuracy by identifying the most significant predictor variables. It helps in eliminating the variables that have little or no impact on the target variable. Regression analysis also helps in identifying outliers and influential observations that can affect the accuracy of the model. By using regression analysis, we can build more accurate predictive models.

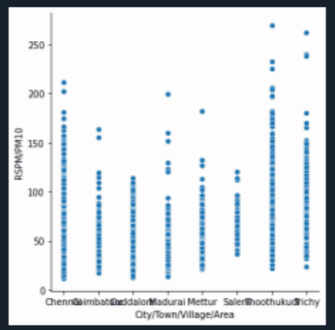
Challenges in Regression Analysis

Regression analysis has some challenges, including multicollinearity, heteroscedasticity, and overfitting. Multicollinearity occurs when two or more independent variables are highly correlated. Heteroscedasticity occurs when the variance of the error term is not constant. Overfitting occurs when the model is too complex and fits the training data too well. These challenges need to be addressed to build accurate predictive models.




File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\balur\.spyder-py3\regression model of rspm.py
temp.py
marginalworkers.py
untitled0.py
regression model of rspm.py*
untitled2.py
#ACCURACY OF PREDICITIVE MODEL USING REGRESSION
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.metrics import r2_score, mean_squared_error
# Load the data
data = pd.read_csv("C:/Users/balur/Downloads/pollution.csv")
# Visualize the data
sns.relplot(x='City/Town/Village/Area', y='RSPM/PM10', data=data)
# Data preprocessing
data1 = data.dropna(subset=['SO2', 'NO2', 'RSPM/PM10'])
x = data1[['SO2', 'NO2']]
y = data1['RSPM/PM10']
# Split the data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
# Create and train a Linear Regression model
linear_regr = LinearRegression()
linear_regr.fit(x_train, y_train)
y_pred_linear = linear_regr.predict(x_test)
# Create and train a Random Forest Regression model
rf_regr = RandomForestRegressor(n_estimators=100, random_state=42)
rf_regr.fit(x_train, y_train)
y_pred_rf = rf_regr.predict(x_test)
# Create and train a Gradient Boosting Regression model


C:\Users\balur\.spyder-py3\regression model of rspm.py
temp.py
marginalworkers.py
untitled0.py
regression model of rspm.py*
untitled2.py
gb_regr = GradientBoostingRegressor(n_estimators=100, random_state=42)
gb_regr.fit(x_train, y_train)
y_pred_gb = gb_regr.predict(x_test)
# Evaluate the models
r_linear = r2_score(y_test, y_pred_linear)
rmse_linear = np.sqrt(mean_squared_error(y_test, y_pred_linear))
r_rf = r2_score(y_test, y_pred_rf)
rmse_rf = np.sqrt(mean_squared_error(y_test, y_pred_rf))
r_gb = r2_score(y_test, y_pred_gb)
rmse_gb = np.sqrt(mean_squared_error(y_test, y_pred_gb))
print ("REGRESSION PREDICTIVE MODEL OF RSPM")
print()
print("Linear Regression - R-squared:", r_linear)
print("Linear Regression - RMSE:", rmse_linear)
print()
print("RANDOM FOREST PREDICITIVE MODEL OF RSPM")
print()
print("Random Forest Regression - R-squared:", r_rf)
print("Random Forest Regression - RMSE:", rmse_rf)
print()
print ("GRADIENT BOOSTING PREDICITVE MODEL OF RSPM")
print()
print("Gradient Boosting Regression - R-squared:", r_gb)



Help | Variable Explorer | Plots | Files

Console 1/A

```
In [1]: runfile('C:/Users/balur/.spyder-py3/regression model of
rspm.py', wdir='C:/Users/balur/.spyder-py3')
REGRESSION PREDICTIVE MODEL OF RSPM

Linear Regression - R-squared: 0.20658507746336507
Linear Regression - RMSE: 28.904650575972003

RANDOM FOREST PREDICITIVE MODEL OF RSPM

Random Forest Regression - R-squared: 0.3088155823904213
Random Forest Regression - RMSE: 26.9782957134439

GRADIENT BOOSTING PREDICITVE MODEL OF RSPM

Gradient Boosting Regression - R-squared: 0.37130960561900983
```

Conclusion

Regression analysis is a powerful tool that can be used to enhance predictive modeling accuracy. It helps in identifying the most significant predictor variables, eliminating variables that have little or no impact, and identifying outliers and influential observations. Despite the challenges, regression analysis is widely used in various fields to make informed decisions based on data. Thank you for your attention.