Predicting Crop Yield in Maji Ndogo:A Simple Linear Regression

Project Overview

This project uses simple linear regression to analyze the relationship between environmental factors and crop yield in Maji Ndogo. The goal is to help local farmers in our fictional country improve their harvests and promote sustainable agriculture. The project includes data exploration, visualizing relationships, fitting linear regression models, and evaluating model performance. Residual analysis is used to assess model fit. Overall, the project aims to provide insights for improving crop yield in the region.

Data Description

The dataset contains various environmental factors collected from agricultural surveys conducted at farms across Maji Ndogo. The factors include elevation, average temperature, pollution levels, slope, and other variables that might influence crop yield.

Key Features of Project

- Examine the dataset to understand the variables and their potential influence on crop yield.
- Visualizing the Relationship between individual factors (e.g., average temperature) and crop yield (Standard_yield).
- Fitting a simple linear regression to quantify the relationship between each factor and crop yield.
- Using fitted regression models to predict future yields based on these factors.
- Analyze the results to determine which factors have a significant impact on crop yield.

Tools Used

- Python
- Jupyter notebooks/VS code/Google Collab
- Pandas library
- Scikit_learn library
- Numpy library
- Scipy library
- Matplotlib and Seaborn libraries
- SQLite Database File

- Data ingestion and Field processor modules (link in ipynb file)

What was done

1. The project began by importing data from an SQLite database into a single DataFrame in Python, creating a comprehensive dataset by joining multiple tables using a single SQL query, and cleaning the data to address issues such as swapped column names, spelling errors in crop types, and negative elevation values using modules created from our last python project (python project 5).

2. With our data cleaned, the next step was to generate a scatter plot to visualize the relationship between Ave_temps and Standard_yield. A Python function called get_correlation was created to calculate the Pearson correlation coefficient between any two numeric columns in the DataFrame. The function takes three parameters: the DataFrame and the names of the two columns for which the correlation coefficient should be calculated.

3. Next, a function fit_linear_regression_model was created that takes a DataFrame and the pollution levels and standard yield columns. This function will fit a linear regression model to the data and return the model object, the model predictions, and the actual y-values. A scatter plot displaying the line of best fit was created to visualize the effect of pollution on standard yield based on the predictions from the linear regression model. The get_correlation function in step 2 was used to calculate the Pearson correlation coefficient between pollution levels and standard yield. This coefficient helps quantify the strength and direction of the linear relationship between the two variables. Finally, a function “get_slope_intercept” was defined to calculate the slope and intercept of the linear regression model to provide insights into how changes in the independent variable (pollution levels) impact the dependent variable (standard yield).

4. A function “calculate_evaluation_metrics” was defined to calculate the R-squared, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) of the linear regression model's performance. This provides an assessment of how well our model fits the data and makes predictions about crop yield based on pollution levels. The function takes the predictions and actual y-values from the model as

input and returns these evaluation metrics as a tuple in the order: R-squared, MAE, MSE, RMSE.

5. A function named “train_split_linear_regression_model” was then defined to train a simple linear regression model on the training set (X_train, y_train), makes predictions on the testing set (X_test), and returns the model, the predictions, and the actual y_test values. Additionally, the function calculates the R-squared, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) using the calculate_evaluation_metrics function (from step 4) and return these evaluation metrics as a tuple. This process helps evaluate the model's performance on unseen data and ensures that it generalizes well.

6. Finally, the fit of the regression model is assessed to help ensure that the model meets the underlying assumptions of linearity, independence, and homoscedasticity of residuals. To begin the analysis, The residuals of the model is calculated by subtracting the predicted values from the actual values in the testing dataset. The distribution of the residuals is plotted using a histogram. A well-fitted model should have residuals that are normally distributed around zero, indicating that the model's predictions are unbiased. Additionally, a scatter plot of the residuals against the predicted values was created to help assess whether the residuals exhibit any patterns or trends. This would suggest that the model is not capturing all the relevant information in the data. The residuals were further analyzed by calculating their mean and standard deviation to provide insights into the spread and central tendency of the residuals. This helps evaluate the overall performance of the model.

Conclusion:

This project aims to provide valuable insights into the factors affecting crop yield in Maji Ndogo, helping local farmers make informed decisions to improve their harvests and promote sustainable agricultural practices.