LOGISTIC REGRESSION : INTERPRETATION  
The logistic regression results you provided can be interpreted as follows:

**Results Interpretation**

1. **Model Coefficients and Intercept**:
   * Model Coefficients:

python

Copiar código

[[ 7.93504080e-09 -1.79724048e-05 -3.34915919e-07 7.26659108e-10]]

These coefficients represent the weights of the independent variables in your logistic regression model. For each independent variable, the coefficient indicates the log-odds of the target variable (price\_change = 1) increasing by one unit.

* + Intercept:

python

Copiar código

[-6.58023375e-07]

This is the bias term in the logistic regression equation.

1. **Accuracy Score**:
   * Accuracy on Test Data: 0.5122338387432263 The model correctly predicted the target variable 51.22% of the time on the test set.
2. **Classification Report**:
   * The classification report provides precision, recall, and F1-score for each class:
     + **Class 0 (price did not increase)**:
       - Precision: 0.53
       - Recall: 0.67
       - F1-score: 0.59
     + **Class 1 (price increased)**:
       - Precision: 0.48
       - Recall: 0.34
       - F1-score: 0.39
   * The **macro average** averages the metrics across classes equally, while the **weighted average** adjusts for the class imbalance.
3. **Performance Observations**:
   * The model is slightly better at predicting when the price does not increase (Class 0) than when it does (Class 1).
   * The low F1-scores, especially for Class 1, indicate room for improvement in model performance, perhaps due to data imbalance, feature selection, or model choice.

**Accuracy Score Computation**

The accuracy score is calculated using accuracy\_score(y\_test, y\_pred):

* y\_test: The true labels for the test set.
* y\_pred: The predicted labels for the test set.

The formula for accuracy:

Uma imagem com texto, Tipo de letra, branco, captura de ecrã

Descrição gerada automaticamenteIn your case, 51.22% of the predictions matched the actual labels. This is a direct result of the accuracy\_score function applied to y\_test and y\_pred.

**Improving regressions: excel data access python (on 07/01/2024)**

> logistic regression:

I have creating a logistic regression (depende variable = stock returns; 1 if price of the stock apreciates, 0 otherwise) and independent variables new\_cases\_smoothed, new\_deths smoothing, dummy variable. But the results were not good, and the variables had lower explanatory value. Give me alternatives of logistics regression, or combinations of variables to improve the explanotry value (like including other variables such us vaccination rate or include interception terms). Hve into account that are variables that have no values in the excel file

**Dito pelo ChatGPT:**

ChatGPT

Improving the explanatory value of your logistic regression model involves selecting meaningful independent variables and accounting for interactions or nonlinear relationships. Here's a structured approach to refine your model:

**1. Feature Engineering**

Feature engineering helps derive more informative variables from the existing dataset.

**Suggestions:**

* **Lagging Variables**:
  + Create lagged variables for **new\_cases\_smoothed** and **new\_deaths\_smoothed** to capture delayed effects on stock performance (e.g., 1-day, 7-day lags).
* **Rolling Averages**:
  + Use rolling averages or cumulative sums of key variables (e.g., 7-day rolling average of **new\_cases\_smoothed**).
* **Ratios**:
  + Compute ratios such as **new\_deaths\_smoothed / new\_cases\_smoothed** to capture mortality trends.
  + Include ratios involving population (e.g., cases per million, deaths per million).
* **Dummy Variable Interactions**:
  + Include interaction terms involving the **Dummy\_Variable** to account for conditions represented by the dummy variable.

**2. Adding Potentially Relevant Variables**

Expand your independent variable set by including other factors that could influence stock returns:

* **Vaccination Rate:**
  + If vaccination data is available, incorporate it as a variable.
  + Create vaccination ratios (e.g., vaccinated population per million).
* **Economic Indicators**:
  + Include variables like **Volume** to capture trading activity.
  + Add variables such as **High** and **Low** to capture volatility.
* **Sentiment and Mobility**:
  + If data is available, include Google mobility trends, consumer sentiment indices, or lockdown status.

**3. Interactions and Polynomial Terms**

Introduce interactions or nonlinear terms to capture complex relationships:

**Examples:**

* **Interaction Terms**:
  + Interaction between **new\_cases\_smoothed** and **new\_deaths\_smoothed**.
  + Interaction between **new\_cases\_smoothed** and the **Dummy\_Variable**.
* **Quadratic Terms**:
  + Include squared terms for continuous variables (e.g., \text{new\_cases\_smoothed}^2) to model diminishing effects.

**4. Addressing Missing Data**

Missing data can reduce the explanatory power of your model. Address this issue through:

* Imputation:
  + Fill missing values with rolling means, medians, or interpolated values.
* Indicator Variables:
  + Create binary indicators to flag rows with missing values.
* Subset Analysis:
  + Restrict your model to rows with complete data for critical variables.

**5. Exploring Alternative Models**

If logistic regression underperforms, consider these alternatives:

* **Regularized Logistic Regression**:
  + Use Lasso (L1) or Ridge (L2) regularization to handle multicollinearity and reduce overfitting.
* **Tree-Based Models**:
  + Decision trees, random forests, or gradient boosting methods like XGBoost provide flexible and interpretable alternatives.
* **Generalized Additive Models (GAMs)**:
  + GAMs allow you to model nonlinear relationships while maintaining interpretability.

**6. Assessing Multicollinearity**

Check for multicollinearity among independent variables using the Variance Inflation Factor (VIF). High VIF values suggest redundancy, which can degrade model performance.

**Example Refinement**

Here’s an example set of independent variables for logistic regression:

* **Lagged\_new\_cases** (7-day lag of new\_cases\_smoothed)
* **Lagged\_new\_deaths** (7-day lag of new\_deaths\_smoothed)
* **Rolling\_avg\_cases** (7-day rolling average of new\_cases\_smoothed)
* **Rolling\_avg\_deaths** (7-day rolling average of new\_deaths\_smoothed)
* **Vaccination\_Rate**
* **Interaction: Dummy\_Variable \* new\_cases\_smoothed**
* **Interaction: new\_cases\_smoothed / total\_cases**
* **Volume**

Would you like assistance in implementing these changes in your dataset or building the revised logistic regression model?

4o

**Dito por si:**

new\_cases\_smoothed and the Dummy\_Variable. --> what would this do? like the interpretation?

**Dito pelo ChatGPT:**

ChatGPT

Including an **interaction term** between new\_cases\_smoothed and Dummy\_Variable in your logistic regression model allows you to explore how the effect of new\_cases\_smoothed on stock returns differs depending on the value of Dummy\_Variable.

**What Does It Mean?**

An interaction term in a logistic regression model is essentially a variable that represents the combined influence of two variables. It helps assess whether the relationship between one independent variable and the dependent variable changes at different levels of another independent variable.

In your case:

* new\_cases\_smoothed represents the smoothed daily number of new COVID-19 cases.
* Dummy\_Variable could represent a specific condition or scenario (e.g., a particular policy, market state, or economic threshold).

By including their interaction term (new\_cases\_smoothed \* Dummy\_Variable), you're testing whether the relationship between new\_cases\_smoothed and the likelihood of stock returns being positive (dependent variable = 1) depends on the value of Dummy\_Variable.

**How to Interpret the Interaction Term**

1. **Positive Coefficient**:
   * If the interaction term has a positive coefficient, it indicates that the impact of new\_cases\_smoothed on stock returns becomes stronger (more positive) when Dummy\_Variable = 1.
2. **Negative Coefficient**:
   * A negative coefficient suggests that the impact of new\_cases\_smoothed on stock returns diminishes (becomes less positive or more negative) when Dummy\_Variable = 1.
3. **Zero Coefficient**:
   * A near-zero coefficient implies no significant change in the relationship between new\_cases\_smoothed and stock returns based on Dummy\_Variable.

**Example**

Suppose your model equation is:

\text{logit}(P(\text{Stock Returns} = 1)) = \beta\_0 + \beta\_1 \cdot \text{new\_cases\_smoothed} + \beta\_2 \cdot \text{Dummy\_Variable} + \beta\_3 \cdot (\text{new\_cases\_smoothed} \cdot \text{Dummy\_Variable})

* **When Dummy\_Variable = 0**: The effect of new\_cases\_smoothed on stock returns is governed by β1\beta\_1β1​ alone.
* **When Dummy\_Variable = 1**: The effect of new\_cases\_smoothed becomes β1+β3\beta\_1 + \beta\_3β1​+β3​.

Here, β3\beta\_3β3​ tells you how much the effect of new\_cases\_smoothed changes when Dummy\_Variable switches from 0 to 1.

**Real-World Example**

Imagine Dummy\_Variable = 1 during lockdown periods and 0 otherwise. The interaction term would test whether the impact of new\_cases\_smoothed on stock returns is different during lockdowns compared to normal periods.

* **Scenario 1**: During lockdown (DummyVariable=1Dummy\_Variable = 1DummyV​ariable=1), high new\_cases\_smoothed may have a stronger negative impact on stock returns due to increased economic concerns.
* **Scenario 2**: During non-lockdown (DummyVariable=0Dummy\_Variable = 0DummyV​ariable=0), new\_cases\_smoothed might have a weaker or even no effect on stock returns.

**Benefits**

* Captures **context-specific effects**: Helps identify whether certain variables' impacts are conditional on other factors.
* Improves model interpretability: Provides insights into how relationships vary across different conditions.

Would you like help constructing this interaction term or interpreting results further?

4o

Parte superior do formulário

Parte inferior do formulário