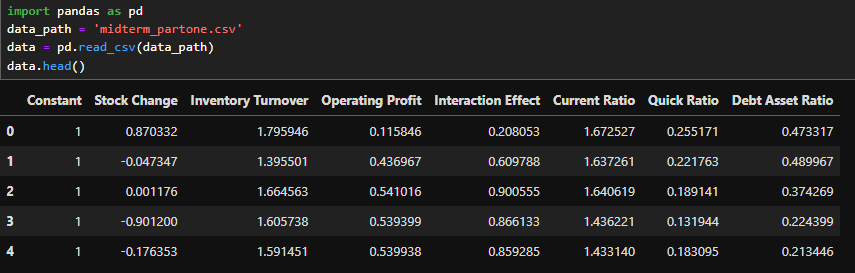
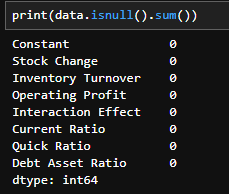
**Part one:**

1. In this part of code we loaded the dataset into dataframe.



1. Here we checked if data have any null values or not.

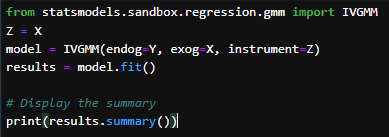


1. The stats models library to perform an instrumental variables estimation using Generalized Method of Moments (GMM) approach.This method is often used in econometrics and statistics to estimate parameters of a model when there are potential issues it meaning that one or more of the explanatory variables are correlated with error term.

The dependent variable is "Stock Change", indicating model aims to explain changes in stock values.

The "Hansen J" statistic and its p-value test model's overidentifying restrictions, which values the validity of instruments. However, p-value is shown as nan (not a number), suggesting there might be an issue with its calculation or interpretation of this statistic.

The coefficients and their significance levels (P>|z|) indicate how each independent variable affects dependent variable. For example, "Operating Profit" has a significant negative effect on "Stock Change" (coefficient = -0.1211, p-value = 0.000), whereas variables like "Inventory Turnover" and "Debt Asset Ratio" have coefficients that are not statistically significant (p-values of 0.972 and 0.981, respectively).



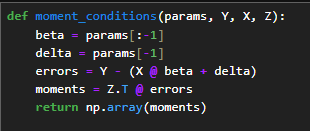
A screenshot of a computer screen

Description automatically generated

1. The code uses scipy.optimize.minimize with L-BFGS-B algorithm damn popular choice for optimizing functions that may not be well-behaved or have large parameter spaces.

initial\_params is an array of zeros to start optimization process, with a length equal to number of coefficients (beta) plus one for delta.

The minimize function seeks to find parameter values that minimize GMM criterion function and effectively estimating the coefficients (beta) and the intercept term (delta) that best fit model given the instrumental variables.



A computer screen shot of a program code

Description automatically generated

**Results:**

It suggesting the best-fitting model under GMM framework considering instrumental variables provided.

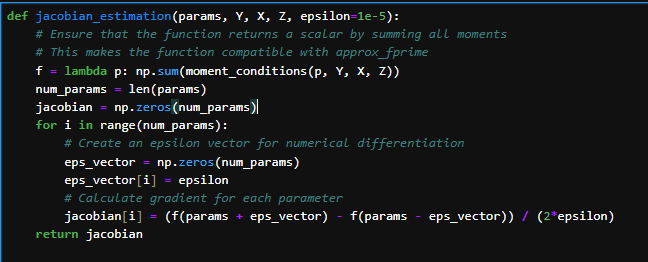
Results showing how each variable is expected to influence dependent variable Y when issues is addressed through instrumental variables in Z.

**Updated Code:**

The optimization is carried out using the minimize function from scipy.optimize, specifying L-BFGS-B method. This method is suitable for problems where the gradient (Jacobian) of objective function is available or can be approximated. The optimization aims to find parameter values (including both the coefficients beta and intercept delta) that minimize GMM criterion.

**Results:**

These values represent estimated impact of each independent variable in X on dependent variable Y. For instance and coefficient of -0.1165653 for a variable suggests that and holding all other variables constant, and one-unit increase in this variable is associated with a decrease of approximately 0.1165653 units in dependent variable Y.

****

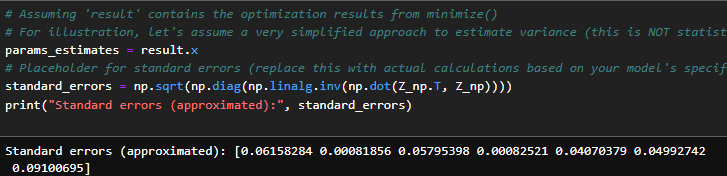
**A computer screen with text and images

Description automatically generated**

**A computer screen with white text

Description automatically generated**

This showing the standard error of all variables.



**Final Results:**

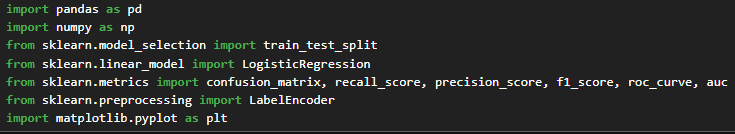
**A computer screen shot of text

Description automatically generated**

There isn't strong statistical evidence to support the claim that there is a notable bias (δ being non-zero) in the instrumental-variable moment conditions based in model and the data analyzed.

**Part 2:**

1. Importing Necessary libraires.



1. Loading the Dataset into dataframe form analyzing and fitting the model.

A screenshot of a computer

Description automatically generated

1. Checking the all columns of dataset.

A screen shot of a computer

Description automatically generated

1. Defining X and Y and converting them into numericals values so it can used in model

A screen shot of a computer code

Description automatically generated

1. Splitting the data into 50% for training and testing.

A computer screen with text and images

Description automatically generated

1. Predicting the values and getting the results (evaluation metrices).

The results suggest that classifier predicted all instances as positive and with a substantial number of False Positives that indicating no True Negatives or False Negatives. This could be indicative of a model with a bias towards predicting positive class or a dataset highly imbalanced towards positive class. The precision indicates that when model predicts a positive result and it is correct approximately 73.48% of time while F1 score suggests a moderate balance between precision and recall.

A computer screen with white text

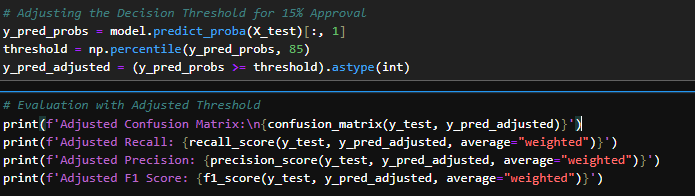
Description automatically generated

1. Setting the decision threshold to adjust and optimize the model

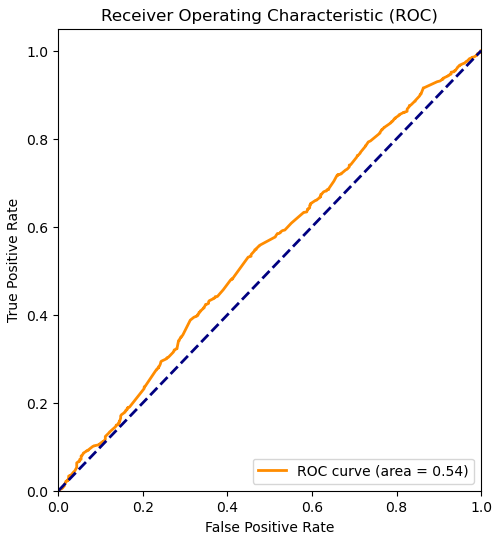
The model has a high precision and meaning it is quite accurate when it predicts an instance to be positive. However this comes at cost of recall that indicating that a large number of positive instances are being missed.

The low recall value suggests that model is conservative in predicting positive instances, potentially leading to a high number of false negatives.

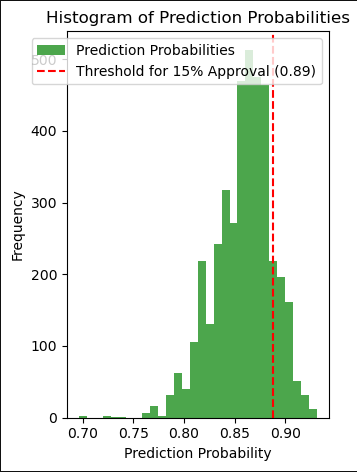
The low F1 score reflects imbalance between precision and recall and indicating that while model is reliable in its positive predictions and it fails to identify a significant portion of positive cases affecting its overall performance in a balanced manner,



**Visualization:**



* The curve is only slightly above line of no skill that indicating that model is not much better than random guessing at distinguishing between two classes.
* The AUC of 0.54 is very close to 0.5 which is not very good and an ideal model would have an AUC much closer to 1.
* Area Under the Curve (AUC): The AUC value is 0.54, as indicated by label "ROC curve (area = 0.54)". This means that model has a 54% probability of distinguishing between the positive and negative class. An AUC of 0.5 suggests no discriminative power (equivalent to random guessing) and while an AUC of 1.0 indicates perfect discrimination.



* The histogram indicates that most prediction probabilities are clustered between approximately 0.75 and 0.85. Very few predictions have probabilities higher than the threshold of 0.89. This threshold seems to be set quite high, suggesting that the model is conservative with respect to what it considers a positive case.
* This kind of threshold setting could be used in scenarios where false positives have a high cost, and thus, a high certainty is required before making a positive classification.