PRODUCT TESTING DOCUMENT

A Purposeful Walk Down Wallstreet

Nabeel Asghar | Michael Shields | Shojib Miah | Michael Chen
Wayne State University – March 23rd, 2020
Senior Capstone Project

Revision History

Date	Description	Author
3/25/2020	Started on document, finished functional test cases.	Nabeel Asghar
3/26/2020	Finished nonfunctional test cases and rest of the document and wrote the test cases.	Nabeel Asghar

Document Approval

Printed Name	Title	Date
Michael Shields	Team Lead	3/29/2020
Nabeel Asghar	Documentation Lead	
Michael Chen	Development Lead	
Shojib Miah	Research and Development Lead	

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this testing document is to show the test plan and test specification combined into one document. This document will showcase the testing of our additions to the existing GM FinTech application. These additions include forecasts based on macroeconomic variables and a polynomial regression algorithm. The test cases will go through functionality, front-end, back-end, and middleware code. The audience this document is intended for our developers and project manager. This document may also be shared with our client or users to showcase functionality and approval. The test cases will be only about the additions to the program that our group made on request of the client.

1.2 References

The test cases that this document has will be front-end based. The front-end is Tableau and the visualization of data. Use cases will also be referenced to show test cases.

2. Functional Testing

2.1 Approach

We will approach the test with a fully working application with all dependencies installed and Tableau working. There will be SQL scripts that will generate tables and get data for data visualization in Tableau.

2.2 Pass / Fail Criteria

The pass grade will be given if the desired result is returned. This test will be considered a success. If the return result is not expected that it will not be a pass. This will be a failed test that must be fixed.

2.3 Entry / Exit Criteria

The criteria for the entry of testing is that Prototype 3 is completed, and the exit criteria is that all functional test have been passed.

2.4 Suspension / Resumption Criteria

If a test case fails, then testing will be suspended, and it will only be resumed when that error is prevented and fixed.

2.5 Risks / Issues

SQL scripts are in danger of dropping entire database to show how they create the tables. The connection to the MySQL server must be concrete and have no issues in it. The computer must have software and hardware requirements as stated in the software requirements documents.

2.6 Expected Results

The expected result for most of the test cases should be output from the console or data value return. Some of the results may simply be data visualization.

2.7 Priority

The priority is based on how essential any test is to the overall application.

2.8 Preconditions and Postconditions

We assume all the dependencies of the application are installed and the MySQL database is setup with the proper username and password in our database engine python file. Since we are going over the additions that we made, we assume all the test cases of the previous groups have passed. The post conditions will depend on the success or fail of the test cases, functional and nonfunctional.

2.9 Test Cases

Test Case ID	TC-0
Test Case Name	All previous test cases are working.
Created By	Nabeel Asghar
Priority	High
Description	Since we inherited this project from previous groups, we need to make sure that all the previous test cases are working. This is everything from the database connection and database entry, to the visualization of data in Tableau.
Preconditions	1. Project is open in PyCharm. 2. MySQL is working. 3. The database connection is working.
Postconditions	1. Test is completed successfully. 2. The rest of the application is working.
Test Steps	N/A
Expected Results	Everything works on the application.
Pass/Fail Criteria	**Pass**: Everything works. **Fail**: Something does not work.

Test Case ID	TC-1.0
Test Case Name	The part of the function macroFetch() which fetches data from Quandl works.
Created By	Nabeel Asghar
Priority	High
Description	The data fetch for macroeconomic variables from Quandl is very important for the rest of the application that we made. All calculations and visualized data are based on these statistics.
Preconditions	1. TC-0. 2. Project is open in PyCharm. 3. MySQL is working. 4. The database connection is working.
Postconditions	1. Test is completed successfully. 2. The data fetched from Quandl is pushed to the database. 3. Database will have accurate values. 4. This data will be visualized in Tableau.
Test Steps	1. Open PyCharm. 2. Navigate to FinsterTab → F2019 → Unit Tests. 3. Run TC2020_Overall.py 4. The test is executed.
Expected Results	Tester should receive the message that the test has passed with the following message: “Test Passed.”
Pass/Fail Criteria	**Pass**: If you receive message “Test Passed” **Fail**: If you received message “Assertion Failed”

Test Case ID	TC-2.0
Test Case Name	The part of the function macroFetch() which fetches data from Fred works.
Created By	Nabeel Asghar
Priority	High
Description	The data fetch for macroeconomic variables from the Fred API is very important for the rest of the application that we made. All calculations and visualized data are based on these statistics.
Preconditions	1. TC-0. 2. Project is open in PyCharm. 3. MySQL is working. 4. The database connection is working
Postconditions	1. Test is completed successfully 2. The data fetched from Fred is pushed to the database 3. Database will have accurate values 4. This data will be visualized in Tableau
Test Steps	1. Open PyCharm. 2. Navigate to FinsterTab → F2019 → Unit Tests 3. Run TC2020_Overall.py 4. The test is executed
Expected Results	Tester should receive the message that the test has passed with the following message: “Test Passed.”
Pass/Fail Criteria	**Pass**: If you receive message “Test Passed” **Fail**: If you received message “Assertion Failed”

Test Case ID	TC-3.0
Test Case Name	The part of the function macroFetch() which fetches TYX data from Yahoo Finance works.
Created By	Nabeel Asghar
Priority	High
Description	The data fetch for macroeconomic variables from the Yahoo Finance API is very important for the rest of the application that we made. All calculations and visualized data are based on these statistics.
Preconditions	1. TC-0. 2. Project is open in PyCharm. 3. MySQL is working. 4. The database connection is working
Postconditions	1. Test is completed successfully 2. The data fetched from Yahoo is pushed to the database 3. Database will have accurate values 4. This data will be visualized in Tableau
Test Steps	1. Open PyCharm. 2. Navigate to FinsterTab → F2019 → Unit Tests 3. Run TC2020_Overall.py 4. The test is executed
Expected Results	Tester should receive the message that the test has passed with the following message: “Test Passed.”
Pass/Fail Criteria	**Pass**: If you receive message “Test Passed” **Fail**: If you received message “Assertion Failed”

Test Case ID	TC-4.0
Test Case Name	The table macroeconmaster exists and it has 8 macroeconomic values in it.
Created By	Nabeel Asghar
Priority	High
Description	It is extremely important that the macroeconmaster table has all of the macroeconomic names as we use this to organize our SQL scripts.
Preconditions	1. TC-0. 2. Project is open in PyCharm. 3. MySQL is working. 4. The database connection is working
Postconditions	1. Test is completed successfully 2. The columns of macroeconmaster has 8 macroeconomic variables present. 3. This data will be used to help organize our Tableau.
Test Steps	1. Open PyCharm. 2. Navigate to FinsterTab → F2019 → Unit Tests 3. Run TC2020_Overall.py 4. The test is executed
Expected Results	Tester should receive the message that the test has passed with the following message: “Test Passed.”
Pass/Fail Criteria	**Pass**: If you receive message “Test Passed” **Fail**: If you received message “Assertion Failed”

Test Case ID	TC-5.0
Test Case Name	The table macroeconstatistics exists and it has values for the 8 macroeconomic variables.
Created By	Nabeel Asghar
Priority	High
Description	It is extremely important that the macroeconstatistics table has all of the macroeconomic values as we use this to calculate the forecasted values.
Preconditions	1. TC-0. 2. Project is open in PyCharm. 3. MySQL is working. 4. The database connection is working
Postconditions	1. Test is completed successfully 2. Macroeconstatistics has values corresponding to all 8 of the macroeconomic variables that are present in the macroeconmaster table. 3. This data will be used to calculate forecast values.
Test Steps	1. Open PyCharm. 2. Navigate to FinsterTab → F2019 → Unit Tests 3. Run TC2020_Overall.py 4. The test is executed
Expected Results	Tester should receive the message that the test has passed with the following message: “Test Passed.”
Pass/Fail Criteria	**Pass**: If you receive message “Test Passed” **Fail**: If you received message “Assertion Failed”

Test Case ID	TC-6.0
Test Case Name	The table macroeconalgorithmforecast exists and it has values for all the algorithm codes
Created By	Nabeel Asghar
Priority	High
Description	It is extremely important that the macroeconalgorithmforecast table has values corresponding to the algorithm as we will chart this in Tableau to show to the client.
Preconditions	1. TC-0. 2. Project is open in PyCharm. 3. MySQL is working. 4. The database connection is working
Postconditions	1. Test is completed successfully 2. macroeconalgorithmforecast has values corresponding to all of the algorithms we used to forecast. 3. This data will be used to visualize in Tableau
Test Steps	1. Open PyCharm. 2. Navigate to FinsterTab → F2019 → Unit Tests 3. Run TC2020_Overall.py 4. The test is executed
Expected Results	Tester should receive the message that the test has passed with the following message: “Test Passed.”
Pass/Fail Criteria	**Pass**: If you receive message “Test Passed” **Fail**: If you received message “Assertion Failed”

Test Case ID	TC-7.0
Test Case Name	The function calculate_MSF1_forecast() runs successfully
Created By	Nabeel Asghar
Priority	Medium
Description	It is important that the MSF1 forecast algorithm works and runs successfully as this is one of our three functions that we created to forecast future stock prices based on macroeconomic variables.
Preconditions	1. TC-0. 2. Project is open in PyCharm. 3. MySQL is working. 4. The database connection is working
Postconditions	1. Test is completed successfully 2. macroeconalgorithmforecast will populate with values with the algorithm code of MSF1. 3. This data will be used to visualize in Tableau.
Test Steps	1. Open PyCharm. 2. Navigate to FinsterTab → F2019 → Unit Tests 3. Run TC2020_Overall.py 4. The test is executed
Expected Results	Tester should receive the message that the test has passed with the following message: “Test Passed.”
Pass/Fail Criteria	**Pass**: If you receive message “Test Passed” **Fail**: If you received message “Assertion Failed”

Test Case ID	TC-8.0
Test Case Name	The function calculate_MSF2_forecast() runs successfully
Created By	Nabeel Asghar
Priority	Medium
Description	It is important that the MSF2 forecast algorithm works and runs successfully as this is one of our three functions that we created to forecast future stock prices based on macroeconomic variables.
Preconditions	1. TC-0. 2. Project is open in PyCharm. 3. MySQL is working. 4. The database connection is working
Postconditions	1. Test is completed successfully 2. macroeconalgorithmforecast will populate with values with the algorithm code of MSF2. 3. This data will be used to visualize in Tableau.
Test Steps	1. Open PyCharm. 2. Navigate to FinsterTab → F2019 → Unit Tests 3. Run TC2020_Overall.py 4. The test is executed
Expected Results	Tester should receive the message that the test has passed with the following message: “Test Passed.”
Pass/Fail Criteria	**Pass**: If you receive message “Test Passed” **Fail**: If you received message “Assertion Failed”

Test Case ID	TC-9.0
Test Case Name	The function calculate_MSF3_forecast() runs successfully
Created By	Nabeel Asghar
Priority	Medium
Description	It is important that the MSF3 forecast algorithm works and runs successfully as this is one of our three functions that we created to forecast future stock prices based on macroeconomic variables.
Preconditions	5. TC-0. 6. Project is open in PyCharm. 7. MySQL is working. 8. The database connection is working
Postconditions	4. Test is completed successfully 5. macroeconalgorithmforecast will populate with values with the algorithm code of MSF3. 6. This data will be used to visualize in Tableau.
Test Steps	5. Open PyCharm. 6. Navigate to FinsterTab → F2019 → Unit Tests 7. Run TC2020_Overall.py 8. The test is executed
Expected Results	Tester should receive the message that the test has passed with the following message: “Test Passed.”
Pass/Fail Criteria	**Pass**: If you receive message “Test Passed” **Fail**: If you received message “Assertion Failed”

Test Case ID	TC-10.0
Test Case Name	The function calc() runs successfully
Created By	Nabeel Asghar
Priority	Medium
Description	This is a helper function of MSF1 which assists the calculation of certain macroeconomic values.
Preconditions	1. TC-0. 2. Project is open in PyCharm. 3. MySQL is working. 4. The database connection is working
Postconditions	1. Test is completed successfully 2. calc() function returns values for MSF1 to use. 3. This data will be used in the calculation of MSF1 forecast.
Test Steps	1. Open PyCharm. 2. Navigate to FinsterTab → F2019 → Unit Tests 3. Run TC2020_Overall.py 4. The test is executed
Expected Results	Tester should receive the message that the test has passed with the following message: “Test Passed.”
Pass/Fail Criteria	**Pass**: If you receive message “Test Passed” **Fail**: If you received message “Assertion Failed”

Test Case ID	TC-11.0
Test Case Name	The table algorithmmaster has the algorithm “regression” in it.
Created By	Nabeel Asghar
Priority	Low
Description	The algorithm master is used to label and make our database easier to understand.
Preconditions	1. TC-0. 2. Project is open in PyCharm. 3. MySQL is working. 4. The database connection is working
Postconditions	1. Test is completed successfully 2. The table algorithmaster has the regression algorithm in it. 3. This will be used to understand our database design and label our data in Tableau
Test Steps	1. Open PyCharm. 2. Navigate to FinsterTab → F2019 → Unit Tests 3. Run TC2020_Overall.py 4. The test is executed
Expected Results	Tester should receive the message that the test has passed with the following message: “Test Passed.”
Pass/Fail Criteria	**Pass**: If you receive message “Test Passed” **Fail**: If you received message “Assertion Failed”

Test Case ID	TC-12.0
Test Case Name	The table algorithmforecast has the forecasted values of all the instruments using regression.
Created By	Nabeel Asghar
Priority	High
Description	The algorithmforecast table has the value of all forecast values based on the algorithm. Since we added regression as one of the algorithms, the forecast table should have forecasted values which were calculated using our regression function.
Preconditions	1. TC-0. 2. Project is open in PyCharm. 3. MySQL is working. 4. The database connection is working
Postconditions	1. Test is completed successfully 2. The table algorithmforecast has the regression algorithm forecasted values in it. 3. These values will be charted and visualized in Tableau.
Test Steps	1. Open PyCharm. 2. Navigate to FinsterTab → F2019 → Unit Tests 3. Run TC2020_Overall.py 4. The test is executed
Expected Results	Tester should receive the message that the test has passed with the following message: “Test Passed.”
Pass/Fail Criteria	**Pass**: If you receive message “Test Passed” **Fail**: If you received message “Assertion Failed”

Test Case ID	TC-13.0
Test Case Name	The function forecast.calculate_regression() runs successfully
Created By	Nabeel Asghar
Priority	High
Description	This is function that calculates forecasted values based on a polynomial regression algorithm. The function must work to completion to populate the database with forecasted values.
Preconditions	1. TC-0. 2. Project is open in PyCharm. 3. MySQL is working. 4. The database connection is working
Postconditions	1. Test is completed successfully 2. forecast.calculate_regression() function returns values to the database 3. This data will be used and visualized in Tableau.
Test Steps	1. Open PyCharm. 2. Navigate to FinsterTab → F2019 → Unit Tests 3. Run TC2020_Overall.py 4. The test is executed
Expected Results	Tester should receive the message that the test has passed with the following message: “Test Passed.”
Pass/Fail Criteria	**Pass**: If you receive message “Test Passed” **Fail**: If you received message “Assertion Failed”

3.Nonfunctional Testing

3.1 Approach

We will approach the nonfunctional testing with a fully working application that has passed all the testcases, has the dependencies installed, and Tableau working. The test will be performed in Tableau as prompted by the application.

3.2 Pass / Fail Criteria

The pass grade will be given if the desired result is returned. This test will be considered a success. If the return result is not expected that it will not be a pass. This will be a failed test that must be fixed.

3.3 Entry / Exit Criteria

The criteria for the entry of testing is that Prototype 3 is completed, and the exit criteria is that all functional test have been passed.

3.4 Suspension / Resumption Criteria

If a test case fails, then testing will be suspended, and it will only be resumed when that error is prevented and fixed.

3.5 Risks / Issues

There should be very little risk to the application. Tableau does not modify data directly and only can read the database. There should be little to no issues testing any of the test cases below.

3.6 Expected Results

The expected results should be that Tableau does not give any error messages and that all the data is displayed without any issue.

3.7 Priority

The priority is based on how essential any test is to the overall application.

3.8 Preconditions and Postconditions

We assume all the dependencies of the application are installed and the MySQL database is setup with the proper username and password in our database engine python file. Since we are going over the additions that we made, we assume all the test cases of the previous groups have passed. The post conditions will depend on the success or fail of the test cases, functional and nonfunctional.

3.9 Test Cases

Test Case ID	TC-14.0
Test Case Name	The Macrovariables sheet works and shows values and with working radio buttons
Created By	Nabeel Asghar
Priority	Medium
Description	This sheet visualizes the macroeconomic variables such as GDP, Inflation Rate, etc. and allows you to switch to display any of them based on the radio buttons on the side.
Preconditions	1. TC-0. 2. MySQL server is running, and all values have populated the database. 3. Tableau is open and connected to the database.
Postconditions	1. Test is completed successfully 2. The sheet displays no errors and the macroeconomic variables are graphed and shown. 3. Radio buttons switch the graph based on the button pressed.
Test Steps	1. Open Tableau application 2. Navigate to FinsterTab → F2019 → Tableau → "**GM_FinTech_Application.twb"** 3. Navigate to the sheet: "Macrovariables" 4. The sheet opens with no errors and displays the macroeconomic variables. 5. The radio button on the side correctly switches between the variables.
Expected Results	The data is displayed, and the radio buttons correctly switch between the macroeconomic variables.
Pass/Fail Criteria	**Pass**: Data is correctly displayed and the radio buttons work. **Fail**: The data may be incorrectly displayed, or the radio buttons do not switch between the variables.

Test Case ID	TC-15.0
Test Case Name	The Macroecon Forecast sheet works and shows the values with working radio buttons.
Created By	Nabeel Asghar
Priority	Medium
Description	This sheet visualizes the stock price and the forecast of stock price based on the algorithms: MSF1. MSF2, and MSF3. The radio buttons allow you to switch to display any of them.
Preconditions	1. TC-0. 2. MySQL server is running, and all values have populated the database. 3. Tableau is open and connected to the database.
Postconditions	1. Test is completed successfully 2. The sheet displays no errors and the sheet shows stock price as well as the forecasted stock price. 3. Radio buttons switch the graph based on the button pressed.
Test Steps	1. Open Tableau application 2. Navigate to FinsterTab → F2019 → Tableau → "**GM_FinTech_Application.twb"** 3. Navigate to the sheet: "Macroecon Forecast" 4. The sheet opens with no errors and displays the forecasted stock price and the current stock price. 5. The radio button on the side correctly switches between the algorithms.
Expected Results	The data is displayed, and the radio buttons correctly switch between the forecast algorithms.
Pass/Fail Criteria	**Pass**: Data is correctly displayed and the radio buttons work. **Fail**: The data may be incorrectly displayed, or the radio buttons do not switch between the variables.

Test Case ID	TC-16.0
Test Case Name	The Dashboard shows the two sheets, Macroecon Forecast and Macrovariables in the same sheet as well as the radio buttons.
Created By	Nabeel Asghar
Priority	Low
Description	This Dashboard visualizes the macroeconomic variables alongside the forecasted values based on the algorithms. The radio buttons that accompany each sheet should also be present and work.
Preconditions	1. TC-0. 2. MySQL server is running, and all values have populated the database. 3. Tableau is open and connected to the database.
Postconditions	1. Test is completed successfully 2. The sheet displays no errors and the sheet shows the two sheets alongside each other. 3. Radio buttons switch the graph based on the button pressed.
Test Steps	1. Open Tableau application 2. Navigate to FinsterTab → F2019 → Tableau → "**GM_FinTech_Application.twb"** 3. Navigate to the sheet: "Dashboard" 4. The sheet opens with no errors and displays the Macroecon Forecast and the Macrovariables sheets alongside each other. 5. The radio button on the side correctly switches the graphs.
Expected Results	The data is displayed, and the radio buttons correctly switch between the values.
Pass/Fail Criteria	**Pass**: Data is correctly displayed and the radio buttons work. **Fail**: The data may be incorrectly displayed, or the radio buttons do not switch between the values.

Test Case ID	TC-17.0
Test Case Name	The Regression sheet work and shows stock price alongside forecasted stock price with radio buttons.
Created By	Nabeel Asghar
Priority	Medium
Description	The sheet visualizes the polynomial regression algorithm. It is important that the graph have two lines: the forecasted price and the actual stock price. The sheet should also have radio buttons on the side that switch between stocks to be shown with the regression algorithm.
Preconditions	1. TC-0. 2. MySQL server is running, and all values have populated the database. 3. Tableau is open and connected to the database.
Postconditions	1. Test is completed successfully 2. The sheet displays no errors and the sheet shows two lines: the forecasted price and the actual stock price. 3. Radio buttons switch the stock shown.
Test Steps	1. Open Tableau application 2. Navigate to FinsterTab → F2019 → Tableau → "**GM_FinTech_Application.twb"** 3. Navigate to the sheet: "Regression" 4. The sheet displays no errors and the sheet shows two lines: the forecasted price and the actual stock price. 5. The radio button on the side correctly switches the stock.
Expected Results	The data is displayed, and the radio buttons correctly switch between the stock.
Pass/Fail Criteria	**Pass**: Data is correctly displayed and the radio buttons work. **Fail**: The data may be incorrectly displayed, or the radio buttons do not switch between the stocks.

Appendix

Appendix A: Testing Scripts

Test Case ID	TC-1.0
Test Case Name	The part of the function macroFetch() which fetches data from Quandl works.
Created By	Nabeel Asghar

```
def test_1_if_Quandl_Variables_Present(self):

    query = 'SELECT * FROM dbo_macroeconmaster WHERE accesssource = "Quandl"'
    result = pd.read_sql_query(query, engine)

    assert result is not None
```

Test Case ID	TC-2.0
Test Case Name	The part of the function macroFetch() which fetches data from Fred works.
Created By	Nabeel Asghar

```
def test_2_if_Fred_Variables_Present(self):

    query = 'SELECT * FROM dbo_macroeconmaster WHERE accesssource = "FRED"'
    result = pd.read_sql_query(query, engine)

    assert result is not None
```

Test Case ID	TC-3.0
Test Case Name	The part of the function macroFetch() which fetches TYX data from Yahoo Finance works.
Created By	Nabeel Asghar

```
def test_3_if_Yahoo_Variables_Present(self):
    query = 'SELECT * FROM dbo_macroeconmaster WHERE accesssource = "Yahoo"'
    result = pd.read_sql_query(query, engine)

    assert result is not None
```

Test Case ID	TC-4.0
Test Case Name	The table macroeconmaster exists and it has 8 macroeconomic values in it.
Created By	Nabeel Asghar

```
def test_4_if_Macromaster_Filled(self):
    query = 'SELECT macroeconcode FROM dbo_macroeconmaster'
    result = pd.read_sql_query(query, engine)

    self.assertEqual(result.size, 8)
```

Test Case ID	TC-5.0
Test Case Name	The table macroeconstatistics exists and it has values for the 8 macroeconomic variables.
Created By	Nabeel Asghar

```
def test_5_if_Macroeconstatisctis_Filled(self):
    query = 'SELECT distinct macroeconcode FROM gmfsp_db.dbo_macroeconstatistics;'
    result = pd.read_sql_query(query, engine)

    self.assertEqual(result.size, 7)
```

Test Case ID	TC-6.0
Test Case Name	The table macroeconalgorithmforecast exists and it has values for all the algorithm codes
Created By	Nabeel Asghar

```
def test_6_if_Macroeconalgorithmforecast_Filled(self):
    query = 'SELECT distinct algorithmcode FROM gmfsp_db.dbo_macroeconalgorithmforecast'
    result = pd.read_sql_query(query, engine)

    self.assertEqual(result.size, 3)
```

Test Case ID	TC-7.0
Test Case Name	The function calculate_MSF1_forecast() runs successfully
Created By	Nabeel Asghar

```
def test_7_if_MSF1_Calculated(self):
    query = 'SELECT algorithmcode FROM gmfsp_db.dbo_macroeconalgorithmforecast ' \
            'WHERE algorithmcode = "MSF1"'
    result = pd.read_sql_query(query, engine)

    assert result is not None
```

Test Case ID	TC-8.0
Test Case Name	The function calculate_MSF2_forecast() runs successfully
Created By	Nabeel Asghar

```
def test_8_if_MSF2_Calculated(self):
    query = 'SELECT algorithmcode FROM gmfsp_db.dbo_macroeconalgorithmforecast ' \
            'WHERE algorithmcode = "MSF2"'
    result = pd.read_sql_query(query, engine)

    assert result is not None
```

Test Case ID	TC-10.0
Test Case Name	The function calc() runs successfully
Created By	Nabeel Asghar

```
def test_10_if_Calc_Function_Works(self):
    query = 'SELECT * FROM gmfsp_db.dbo_macroeconalgorithmforecast'
    result = pd.read_sql_query(query, engine)

    assert result is not None
```

Test Case ID	TC-11.0
Test Case Name	The table algorithmmaster has the algorithm “regression” in it.
Created By	Nabeel Asghar

```
def test_11_if_Regression_Present_In_Master(self):
    query = 'SELECT * FROM gmfsp_db.dbo_algorithmmaster ' \
            'where algorithmcode = "regression"'
    result = pd.read_sql_query(query, engine)

    assert result is not None
```

Test Case ID	TC-12.0
Test Case Name	The table algorithmforecast has the forecasted values of all the instruments using regression.
Created By	Nabeel Asghar

```
def test_12_if_Regression_Forecasted_Values_Present(self):
    query = 'SELECT * FROM gmfsp_db.dbo_algorithmforecast ' \
            'WHERE algorithmcode = "regression"'
    result = pd.read_sql_query(query, engine)

    assert result is not None
```

Test Case ID	TC-13.0
Test Case Name	The function forecast.calculate_regression() runs successfully
Created By	Nabeel Asghar

```
def test_13_if_Regression_Function_Runs(self):
    query = 'SELECT instrumentid, algorithmcode FROM ' \
            'gmfsp_db.dbo_algorithmforecast ' \
            'WHERE algorithmcode = "regression"'
    result = pd.read_sql_query(query, engine)

    assert result is not None
```