Repository and Mining of Temporal Data
Milestone 3 Evaluation

Team Members:

Jessica Nguy	jnguy2014@my.fit.edu
Siomara Nieves	snieves2014@my.fit.edu

Faculty Sponsor and Client:

Philip Chan	pkc@fit.edu

Meetings With the Client:
Nov. 13, Nov. 27

Progress of Current Milestone:

Task	**Completion %**	**Jessica**	**Siomara**	**To-Do**
Q2	90%	90%	0%	Django Integration
NarrowData	70%	70%	0%	Refine method of NarrowData, finish parsing and ensure integration works
Database Processing	80%	50%	30%	Populate database with uploaded files on website
Meta-Data, Meta-Data input	70%	50%	20%	Link to Q2 on website.
Improve existing code	95%	80%	15%	Time handling
Website Q1	85%	0%	85%	Redirect to Q1 app.
Evaluation Document, Presentation	100%	50%	50%	None

Discussion of Each Accomplished Task:

Task 1: Q2 parses the initial csv file. The code then retrieves three other csv files, determined by the user and parses them into readable formats. In Milestone 4 this section will be changed to call from the database rather than by the user, and the number of other csv files called will increase. Currently the section that parses the other csv files only accepts csvs that have the same start date and number of items in the list. The shuffle section is called five times per each csv file analyzed. This shuffle is meant to ensure that lag can be calculated against each value of the initial csv file.

Two multidimensional arrays are used to keep track of analyzed csvs, their lags, and information. This is to keep track of the data when visualizing it. Once all the csvs and their lags have been analyzed and inserted into the array, they are then sorted from highest correlation to lowest. The top three csv files, each of which are different (i.e., if NASDAQ at lag=1 is the number 1 in regards to correlation, and NASDAQ at lag=0 is the second, NASDAQ at lag=0 cannot be used.) are selected and graphed.

The first Figure is a line graph of the target variable and the top-3 csv files. The z-score of the variables have not been graphed, but in Milestone 4 this will be fixed. Figure 2 is the amount of correlation of top-k variables against the target variable. The target variable is displayed at the top with correlation=1, and the variables with their lags and amount of correlation is displayed on the bar graph. Figure 3 is a bar graph of the top-1 variable and all of its correlation with lag=0 to lag=5. This graph explains the reasoning why the top-1 variable was selected at that specific lag.

Task 2: Half of NarrowData is done in Q2. This will be remedied in Milestone 4, where the shuffle function is moved into NarrowData.py, and will also shuffle back rather than forwards. Shuffling forwards deletes data points, whereas shuffling back - setting the beginning of the csv files to be compared in timestamps several lags earlier than the start of the initial csv file preserves the number of data points able to be compared. Due to time constraints and lack of knowledge half of NarrowData is incomplete.

Task 3: Database setup is complete using Django's default SQLite and has yet to be populated with the files' metadata inputs on Django. The task includes having the user input the metadata on the website and saving it on the database in order to later be used in NarrowData for Q2. Another database has been created and populated with meta-data from the test csvs that were used to test the code before adding them to the actual Django website; this external database can also be linked to the Django application.

The second database takes in user inputs for file name (the uploaded csv file), a description of the file, the type of granularity the file has (whether or not it is monthly, daily, yearly, etc), and four tags determined by the user. This database automatically calculates additional information for the tables, including start-time and end-time of the dates, the type of privacy, number of items in the list, and the ID number. The ID number is present in all four tables and is the primary key of the entire database. The number will be used to compare values across all the tables to optimize the amount of time the database will take to search through the contents. Currently the privacy is set to ‘admin’ for user and ‘public’ for setting; this will change in future Milestones as we start to use a Username setting.

Task 4: User meta-data input includes time range, additional search values, and a short description of the uploaded csv file which all will be stored in the SQLite database. The meta-data will be used for NarrowData in order to answer Q2 and display the question’s graphs on the website. There are four tables that the meta-data is organized into; description, granularity, privacy, and tags. Each of the tables has at minimum of two different sections of data. All four tables has the same primary key that is labeled IDnum which is generated by the code when Input.py is run. This is to simulate when a user uploads a csv file during Q1. Integration between Input.py and Q1.py or Q2.py does not exist at this point in time.

Task 5: During this task improves include new catch cases for uploading files such as the file not being in .csv format and the file being too big. If there is an error while uploading the file, the website redirects to the upload landing page and reflects an error on why it has failed; also, the file reading process now allows files being in rows rather than columns.

In Q1 the code has been cleaned up to look less cluttered and for ease of reading. The titles of the x-axis, y-axis, and titles of the graph have been improved. Declarations of variables have been decreased. Q1 has been improved as the client has requested; Figures 3 and 4 of Q1 now display as bar graphs rather than line graphs to focus on the change between each point. Figure 4 plots z-scores of -3.0, -2.5, 2.5, and 3.0 with dashed lines; 2.5 (absolute value) are yellow, and 3.0 (absolute value) are red. When the bar graph crosses these thresholds the colors of the line change. Between absolute values of 0.0 and 2.5 the line is green; between 2.5 and 3.0 is yellow; and 3.0 and beyond is red.

Task 6: The website has been setup in a Django application on a local server, the website includes the apps for the main page, the uploading page, and the display of Q1. The main page includes redirecting the user whether it is a provider or a customer, as of now the website is only supporting providers uploading their files. After selecting the option for provider, the website redirects the user to a login page where they can access their account or register, future improvements include providing proof that the providers are reliable users that are allowed to upload the data. Once the provider has logged in, the upload page displays the option for uploading the file; the implementation of asking for the file meta-data will be further discussed in the next milestone. Once the file has been selected and uploaded, the file is checked for errors such as not being a .csv file, the file being too big, and the file containing errors in their data;

after the file has been successfully read, the page redirects to Q1's app displaying the graph. As of now the overall design of the website has been deferred to a later milestone once all three project questions have been answered.

Task 7: The evaluation document was written collectively by using Google Docs and the presentation by using Google Slides.

Discussion of Contribution for Each Task:

Jessica: Fixed Client's request of changing the appearance of Figure 3 and 4 on Q1. Added color to Figure 4 of Q1. Worked on Q2, which is complete aside from setting up on the web application. NarrowData is half-complete. Input.py is mostly complete and is the section that parses user inputs, such as determining the target variable and search tags. Upload.py is meant to imitate when the user uploads a file and asks the user for the file name, a description of the file, granularity settings, and tags to describe the file. Setting up the sqlite3 database to take in the user-defined items from Upload.py and adding them to the database in their required sections. Additional analyzing of the file is done here to add additional information to the database. Also contributed to writing the Milestone 3 Evaluation, worked on the Milestone 3 presentation.

Siomara: added meta-data options into Django's pre-defined SQLite database, added meta-data views for website during file upload which are all user input, changed files' model in Django for meta-data handling, improved file reading/upload with new test cases, wrote home page, provider view, file upload view, user registration on website, added a navigation bar and search bar for homepage. Wrote Q1's app and view but need to be redirected after the file upload as well as worked on timestamp parsing which is not complete yet; worked on Milestone 3 document and presentation.

Plan for the Next Milestone:

Task	**Jessica**	**Siomara**
Q3	Brainstorm, Pseudocode, Research if needed, Program Q3.	Q3 app on Django, research, testing
NarrowData	Finish NarrowData; Figure out how to handle multiple checks for csv files	Link to Django

Django Website	Research methods of hosting code; Not on LocalHost (AWS/PythonAnywhere/etc)	Create app for Q3 on current website and redirect after answering Q1 and Q2 Provider Login/logout, Customer view
Save File Uploads	Find more csv files to populate database	Save files locally on a directory for testing, more research on saving files elsewhere
Database input handling	Refine Database Processing to be faster	Django database handling for meta-data input from user for Q2
Improve current code	Z-scores for Q2 Fig1	Test cases for website
Project Plan	Write Plan	Write Plan
Evaluation Document, Presentation	Write evaluation document, Create presentation, put code onto GitHub repository.	Write evaluation document, Create presentation, put code onto GitHub repository.

Discussion of Each Planned Task:

Q3: Split into two sections; Winter Break and January. During Winter Break brainstorm and write pseudocode for Q3; research methods and tools if needed. During January write, implement, and test code for Q3. Implementing Q3 on the current Django website.

NarrowData: Research and brainstorm a method to keep all data that has the same granularity and timestamps in one current place. Data that has a start time that occurs at least 5 timestamps before the uploaded code is kept in an array; shuffle and roll functions have already been written. Checking for tags and target variables needs to be figured out and have pseudocode written out and implemented. Make sure that the code works with Q1.py, Q2.py, Q3.py, and uses calls from Input.py, Upload.py, and parseCSV.py. Test and upload onto GitHub repository.

Django Website: Once Q3 is done, implement a new application on the Django website and show the visualizations after the provider has uploaded the file and after Q1 and Q2 have been shown as well. Create a login/logout option for the provider after they have already registered for an account and begin creating a customer view (customer is not allowed to upload files); customer's view include searching for file data with the already implemented search section on the top navigation navigation bar.

Save File Uploads: Figure out and implement how to save files that are uploaded by a user to the Django application. Search for different .csv files with different granularity and save them locally. Research about saving the files on a different interface (such as Amazon S3). Test and implement the change.

Database Input Handling: Link external database to django and/or implement it on the web application and test if database is saving the meta-data input on website for answering the project's Q2. Add column to description table that reads whether or not the data is in column format or in row format for ease of parsing. If necessary, re-create database to allow for this change.

Improve Current Code: Change visualizations as needed if Faculty Sponsor requests it. During Winter Break have a complete overhaul of existing code to ensure that there is no redundancy. Move NarrowData sections in Q2.py to NarrowData.py. Improve timestamp parsing to different options (monthly, hourly, etc) as well as datetimes shown on visualizations.

Project Plan: Create a Project Plan detailing the second semester of Senior Design, and what is needed/expected during the second phase of the project. The Semester 2 Project Plan is due in the beginning of January; work and complete this section during Winter Break.

Evaluation Document, Presentation: Creation of an evaluation document to indicate what was completed in Milestone 4, future requirements in Milestone 5. Create detailed explanations of completed milestones and detailed plans of future milestones. Create a Presentation for Milestone 4. Upload complete and/or current documents/code to GitHub repository and update links on main page. Write the Milestone 4 Evaluation and create the Presentation before the Milestone 4 due date.

Sponsor Feedback:

Task 1: Different data sets. e.g. unemployment data, GDP data

Task 2: Come up with test cases to test "narrowing"

Task 3: Linear regression

$$y = w_0 + w_1 x_1 + w_2 x_2 + \ldots$$

Find API to generate the model from data

11/27	11/26 or earlier
11/26	11/25 or earlier

Using the model for forecasting

11/28	11/27 or earlier

Task 4:

Task 5:

Task 6:

Task 7:

Task 8:

Sponsor Signature: ____ Date: ___________

Sponsor Evaluation

- Sponsor: detach and return this page to Dr. Chan (HC 322)
- Score (0-10) for each member: circle a score (or circle two adjacent scores for .25 or write down a real number between 0 and 10)

Jessica	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Siomara	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

Sponsor Signature: ____ Date: ___________