Assignment #5

Due Date: 4/21/19 by 11:59pm

Deliverable:

Post your homework as a SINGLE ZIP file on Blackboard with the name "HW5_YourLastName" that has the following:

1. Your updated IPYNB scripts; the entire python scripts that have your source code and the output
2. Your PDF files for your IPYNB scripts
3. 5 minutes video recording using screencast-o-matic (https://screencast-o-matic.com/) to demonstrate the run of your scripts and your output for the ipynb scripts

Important Notes:

- Direct ALL QUESTIONS related to the assignment to the
- Do NOT communicate or share your assignment with others
- Do NOT share your personal laptop with your classmates

High-Level Requirements:

1. The assignment will be completed in 3 phases.
2. All your deliverables will be based on work you have done on GitHub.
3. Phase-I is due on 4/3/2019 by 11:59pm
4. Phase-II is due on 4/9/2019 by 11:59pm
5. Phase-III is due on 4/21/2019 by 11:59pm

Phase-I Requirements:

1. You have to create your GitHub account id as instructed in the tutorial.

- For a student with the name John Doe with A number A12345678.
- His id will be JSP19SCM78D.
- In this ID J is the first character of his first name.
- SP19SCM will be common for each student.
- 78 is the last two numbers of his A number.
- D is the first character of his last name.

Once the TA receives all of your GitHub accounts you will be added as Collaborators to the master repository. For this, the TA will be sending out a mail to get all your created GitHub accounts. Use your hawk id while registering for your GitHub accounts. Do not use newly created email ids just for the assignment as GitHub may consider these id's as Bots and remove them. You will be proceeding to the next step only after the TA sends out a mail confirming that you have been added as a collaborator.

2. Next you will be forking the repository SCM587SP19 from SPM587SP2018 GitHub organization to the GitHub account that you have created. This repository can be opened with the following URL.
   https://github.com/SPM587SP19/SCM587SP19.git

3. Now using the URL given above you will login to the repository listed above. You will go to the issues tab and create a new issue.
4. The **title** of the issue will be as listed below.
   "<Your GitHub id> adding first name"
   Example – "JSP19SCM78D adding first name"
5. This issue will be **labeled** using the labeling scheme listed below.

6. **Labeling Scheme**: Your Labeling Scheme follows the **key:value** format "**LabelName:*LabelValue***" for every label

7. **Types of Labels**: There are different types of labels, the following is the list of labels used for issues listed that you will use based on the **first digit of your A-Number**:
   i. **OriginationPhase**:

1. **If the first digit is odd number:** use one of the values {Coding, Testing, Documentation, Field}
2. **If the first digit is even number:** use one of the values {Requirements, Design }

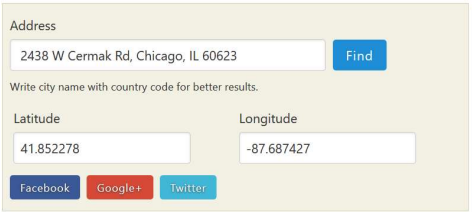
ii. **DetectionPhase**:
1. **If the first digit is odd number:** use one of the values { Coding }
2. **If the first digit is even number:** use one of the values {Design}

iii. **Priority**:
1. **If the first digit is odd number:** use one of the values { Major , Low}
2. **If the first digit is even number:** use one of the values {Medium, Critical}

iv. **Status**:
1. **If the first digit is odd number:** use one of the values { Rejected, Completed }
2. **If the first digit is even number:** use one of the values { pendingReview , Approved, inProgress, }

v. **Category**:
1. **If the first digit is odd number:** use one of the values { Enhancement, Inquiry }
2. **If the first digit is even number:** use one of the values {Bug }

vi. **Address (You need 3 labels for Address, Latitude, and Longitude)**:
1. **If the first digit is odd number:** Use any McDonalds location that is in a zip code that ends up with odd number. For example, the student with A12345678 will google Mcdonalds locations and might pick this address (2438 W Cermak Rd, Chicago, IL 60623). And then uses this URL (https://www.latlong.net/convert-address-to-lat-long.html) to find the Lat and Long coordinate values:



2. **If the first digit is even number:** Use any McDonalds location that is in a zip code that ends up with even number

8. You will now clone the repository that you have forked into your account on to your local machine. This repository will contain files a.java, b.java, c.java, …, z.java. You will open the file that has the first

character of your first name. Once you open the file you will enter your GitHub account id in a new line and save it.

Example – For John Doe with GitHub account JSP19SCM78D, he will open j.java and add his GitHub account username in a new line and save it. So j.java will contain JSP19SCM78D.

9. Now you will do a commit and then push this repository into your GitHub account. While making the commit you have to add your Issue number in the commit message in the following format.

   Example – git commit –m "Fixes #<IssueNumber>. <GitHub username> first name added."

   Say for John Doe the issue he has created is Issue #24. His commit message will be as follows.

   git commit –m "Fixes #24. JSP19SCM78D first name added"

10. Once that is done you will create a pull request to the master repository so that your changes can be merged onto the master repository. You will just create the pull request. It will be the TA who will accept your changes. Your pull request title should also be the exact same as your commit message.

    This will complete your Phase 1. You will also need to submit two screenshots as the commit report for each phase. The last page in this document instructs you on how to capture those screen shots. Both the screenshots that you have captured need to be put in a pdf called Report_Phase1_<GitHubUsername>.pdf for your phase I. Once your changes are all accepted in the master repository you will receive an email to commence Phase II. Then you will be following the Phase II instructions. Do not start Phase II till you have received the email to proceed with Phase II from the TA.

Phase-II Requirements:

1. You will open the master repository from your GitHub account and create a new issue.
2. This issues **title** will be as listed below.
   "<Your GitHub id> adding last name"
   Example – "JSP19SCM78D adding last name"
3. This issue will be **labeled** using the labeling scheme listed below.
4. **Labeling Scheme**: Your Labeling Scheme follows the **key:value** format "**LabelName:*LabelValue***" for every label
5. **Types of Labels**: There are different types of labels, the following is the list of labels used for issues listed that you will use based on the **last digit of your A-Number**:
   i. **OriginationPhase**:
      1. **If the last digit is odd number:** use one of the values {Requirements, Design }
      2. **If the last digit is even number:** use one of the values {Coding, Testing, Documentation, Field}
   ii. **DetectionPhase**:
      1. **If the last digit is odd number:** use one of the values { Testing }
      2. **If the last digit is even number:** use one of the values {Field}
   iii. **Priority**:
      1. **If the last digit is odd number:** use one of the values {Critical, Major }
      2. **If the last digit is even number:** use one of the values { High, Low, Medium}
   iv. **Status**:
      1. **If the last digit is odd number:** use one of the values { Approved, Rejected, Completed }
      2. **If the last digit is even number:** use one of the values { inProgress, pendingReview }
   v. **Category**:
      1. **If the last digit is odd number:** use one of the values { Bug, Enhancement }
      2. **If the last digit is even number:** use one of the values {Inquiry }

6. Now you go to the forked repository in your account and update the repository. This is done using the following command.

   git remote add upstream <Master Repository Git URL>
   git fetch upstream
   git rebase upstream/master

7. Update your local clone to the latest in your repository.

8. Now you will open the file with the first character of your last name. Then you will be adding your GitHub account username in a new line

   Example – John Doe will be opening the file d.java and add his GitHub account username which is JSP19SCM78D in a new line.

9. Now you will do a commit and then push this repository into your GitHub account. While making the commit you are to add your Issue number in the commit message in the following format.

   Example – git commit –m “Fixes #<IssueNumber>. <GitHub username> first name added.”

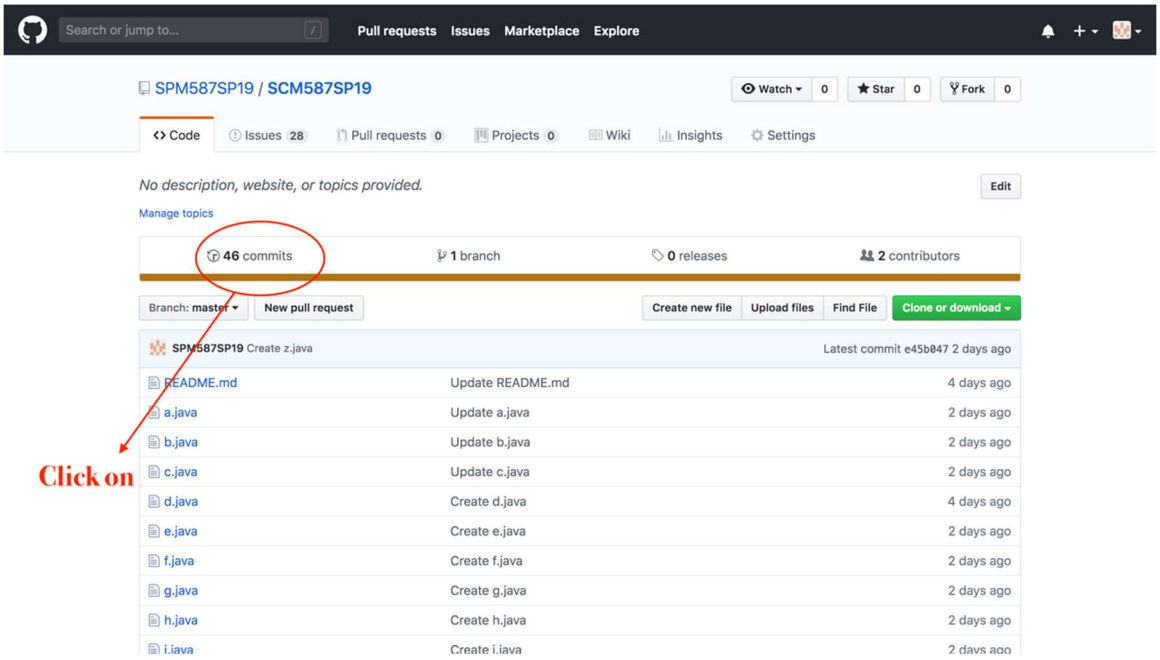
   Say for John Doe the issue he has created is Issue #65. His commit message will be as follows.
   git commit –m “Fixes #65. JSP19SCM78D last name added”

10. Once that is done you will create a pull request to the master repository so that your changes can be merged onto the master repository. You will just create the pull request. It will be the TA who will accept your changes. Your pull request title should also be the exact same as your commit message.

    This will complete your Phase II. You have to capture the same screenshots mentioned above for phase II and attach it in a document called Report_PhaseII_<GitHubUsername>.pdf and submit it.

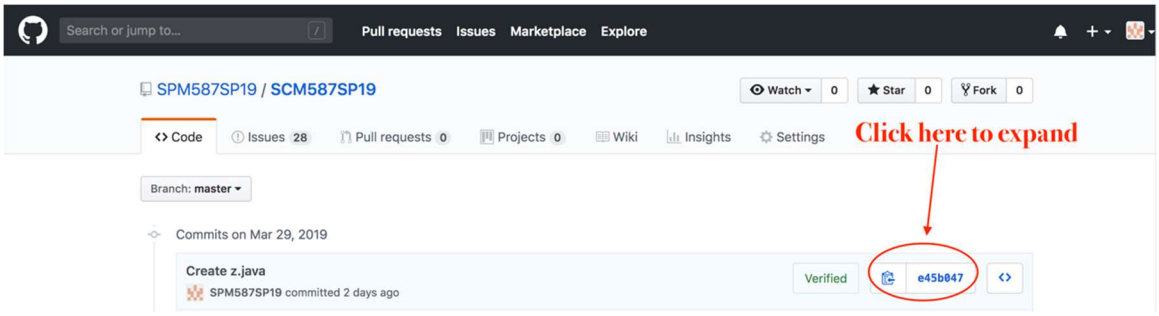
Deliverable for Phase-II :

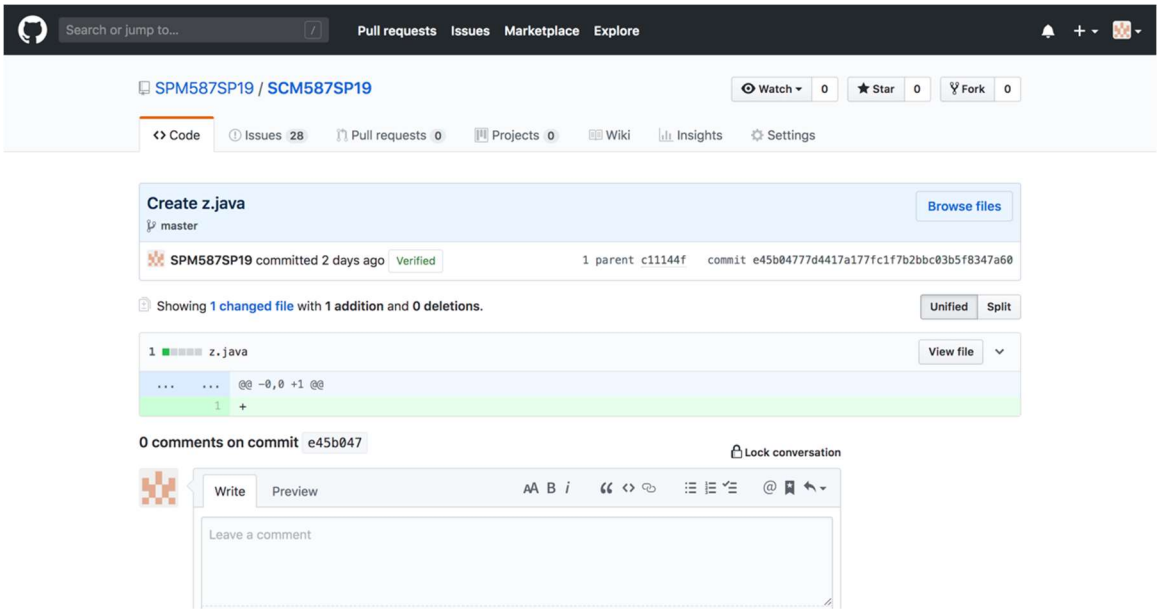
1. Login to your GitHub account and open the repository that you have committed to. Click on the commits Tab.



2. You will get the following page. You are to take a screenshot of this page. Click on the commit that you have just made to expand it.

3. This will expand and show you the individual commit. You have to take a screenshot of this page as well.





The report that you have created will be added to the report folder in the repository and submitted. This means that for the phase I report, you will be adding the report document in that Phase1_Report folder and committing it. And for Phase II you will be adding the report document in the folder Phase2_Report folder and committing it. Make sure that the report for each phase has to be committed and pushed to your repository BEFORE you initiate a pull request.

Phase-III Requirements:

Before you start working on this Phase you must read this article:
https://towardsdatascience.com/time-series-in-python-exponential-smoothing-and-arima-processes-2c67f2a52788

Update and run the modified Python ipynb scripts to do the following:

1. Requirement #1: Plot in Bar Chart the total number of issues closed every day for every Origination Phase
2. Requirement #2: Plot in Bar Chart the total number of issues created for every Phase based on their Status
3. Requirement #3: Plot in Bar Chart the total number of issues for
    1) DetectionPhase is Field AND Priority is Critical
    2) DetectionPhase is Field AND Status is Completed
    3) DetectionPhase is Field AND Priority is Critical AND Status is Approved
    4) DetectionPhase is Field AND Priority is Critical or High AND Status is Approved or inProgress
4. Use Facebook/Prophet package (**https://facebook.github.io/prophet/docs/quick_start.html**) to forecast the following for our repo
    1. The day of the week maximum number of issues created
    2. The day of the week maximum number of issues closed
    3. Plot the created issues forecast by calling the Prophet.plot method and passing in your forecast dataframe.
    4. Plot the closed issues forecast; use the Prophet.plot_components method. By default you'll see the trend, yearly seasonality, and weekly seasonality of the time series. If you include holidays, you'll see those here, too.
    5. Plot the pulls forecast; use the Prophet.plot_components method. By default you'll see the trend, yearly seasonality, and weekly seasonality of the time series. If you include holidays, you'll see those here, too.
    6. Plot the commits forecast; use the Prophet.plot_components method. By default you'll

see the trend, yearly seasonality, and weekly seasonality of the time series. If you include holidays, you'll see those here, too.

5. Re-implement the above 6 requirements (listed for Facebook prophet package) using **TensorFlow Time Series (TFTS)** **https://github.com/tensorflow/tensorflow/tree/master/tensorflow/contrib/timeseries**
6. Re-implement the above 6 requirements (listed for Facebook prophet package) using **StatsModel :** **https://www.statsmodels.org/stable/index.html**

Deliverable for Phase-III :

Submit a compressed ZIP file that has all scripts and output files.
Your ipynb scripts must have the following:

1. Your added code **in every cell** in the script
2. Your output **below every cell** in the script