**Milestone 1**

**Private GitHub Repository for the project(with me as added collaborator):** <https://github.com/msichterman/csce-496-project>

**Group Name with Members:**

Imperial Engineers — Matt Sichterman, Easton Joachimsen, Colin Cummings

**Project Title:**

Travis, Why Do the Builds Keep Failing?: An Analysis on CI Failure Causes and Severity

**Project Abstract:**

Continuous Integration is a practice that has become widely used and is a standard in the industry. Travis CI can be dropped into any GitHub repository and automate various tasks including the build and test processes. With that much convenience at face value, there also comes hiccups when builds fail which can get tedious. This paper strives to quantify the different types and reasons that builds fail and how quickly a successful subsequent build is completed. Using TravisTorrent’s dataset, we apply various data analysis and data visualization techniques to clearly compare and portray failing builds to understand their cause and severity based on the time to the next successful build. Our results suggest that Travis CI fails due to minor blunders most often, specifically, 80% of the time. Also, our results propose that in only 5% of cases do issues cause a severe build failure, which we concluded is any downtime of 24 hours or more.

**Project Motivation (Why is this important to study?):**

The motivation of this project to better understand CI practices and what causes failures within the pipelines. With the simplicity of CI handling the build and test processes automatically within a GitHub repository, it is no surprise that it has become an industry standard and a best practice of modern software development. Due to CI being commonly used, it is important to study more in depth to fully understand how useful the practice really is. We strive to study the cause of failures within the TravisTorrent dataset and seek to quantify their severity based on the time to the next successful build.

**Related Work (At least 3 papers in related work reviewed with a few lines on what the papers analyze. You could use the visual template for these papers if you wanted or just write a paragraph about them):**

1. [***Oops, My Tests Broke the Build: An Explorative Analysis of Travis CI with GitHub***](https://ieeexplore.ieee.org/document/7962385)
   1. This paper is a breakdown on how unit tests impact the way that CI is integrated and analyzes different factors on why testing is a central part of CI. The paper analyzes data from github projects that use travis CI and breaks down a number of different factors such as amount of unit test and languages used in the projects along with build information.
2. [***How Open Source Projects use Static Code Analysis Tools in Continuous Integration Pipelines***](https://ieeexplore.ieee.org/document/7962383)
   1. This paper is a breakdown on how static code analyzers impact the way that CI is integrated into projects and looks into different aspects of static code analyzers such as which tools are being used for static code analysis, which kind of issues that cause warnings to arise or builds to fail, and also how long it takes for different projects to resolve the builds that fail due to these static code analyzers.
3. [***How Does Contributors’ Involvement Influence the Build Status of an Open-Source Software Project?***](https://ieeexplore.ieee.org/document/7962400)
   1. This paper is a high level analysis on open source software projects that use Travis CI. Their primary motivation for their paper was to group contributors to the project into different categories based on the amount and quality of contributions made and to then compare the build statistics gathered for each category. By doing this the paper hoped to achieve a heightened insight into whether or not casual contributors had a higher percentage of build failures compared to more adept developers within the projects.

**Research Question(s):**

* On average what types of errors cause builds to fail the most?
* Does the frequency of commits from a specific user affect build success?
* Do factors such as the size of a commit or the number of jobs affect build success?
* What are the most common types of minor errors(take less than 1 hour to fix) that cause the build to fail?

**Hypotheses (if any):**

* A vast majority of build failures are caused by unit tests failing and minor code bugs, which we expect should take less than an hour to resolve. We predict that failures will be resolved in under an hour 80 percent of the time, and under 5 percent of cases of failing builds will result in a resolvement time of over 24 hours.

**Methods to be Used:**

We will use data analysis techniques to form models of the data in order to formulate answers to our research questions and hypothesis. With that data, we will visualize the results in order to clearly portray our findings. We plan to use Python with the [scikit-learn library](https://scikit-learn.org/stable/) to analyze the data within the TravisTorrent dataset. In order to achieve this, techniques such as data preprocessing, dimensionality reduction, and clustering will be utilized.

**Data to be Used / Obtained:**

We are using the TravisTorrent dataset from the MSR 2017 challenge. We are accessing it through Google BigQuery, which can be accessed [here](https://travistorrent.testroots.org/page_dataformat/). The dataset includes information about the data such as test density, number of test cases and lines modified/deleted/added, test successes/failures, timestamps, and information about the github pull requests and builds. The dataset synthesises data from three different sources being the git repositories, Travis CI’s API and build log, and Github through GHTorrent.

**Analysis Envisioned:**

We will use data science techniques to run data analysis and create data visualizations to support our hypothesis. We plan to use Python with the [scikit-learn library](https://scikit-learn.org/stable/) to analyze the data within the TravisTorrent dataset. In order to achieve this, techniques such as data preprocessing, dimensionality reduction, and clustering will be utilized.

**Possible Threats to Validity Envisioned:**

A possible threat to validity could be the use of partial, or incomplete datasets, misunderstanding of the data pulled from the datasets or even incorrect data queries on the datasets.

**Contributions Envisioned:**

Through the research conducted through this project we hope to contribute an accurate analysis to the CI community that clearly portrays the most frequent cause for build failures and also insight on the time requirement for resolvement.