Thank you for reviewing our proposal for the Concrete Strength Project. Our goal is to code a solution to your problem that allows the user to input their different amount of materials for each mixture. After the user has to input everything, the code will output the concrete strengths for each input. This will shorten the time of waiting for the 28-day approach to just a few minutes that will produce the same results as if you were to use the 28-day approach.

In the graph below, we have provided a timeline for the amount of time it will take to complete this solution.



In the graph, we have lists of all the tasks we plan to do, and a timeline for the amount of time we will spend on each task. The tasks are to be explained in further detail in this report, to provide a better understanding of how each task is important.

The database provided to you is an excel with nine headers, consisting of cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate, age, and concrete compressive strength. The cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate are all predictors for the concrete compressive strength. These are the predictors because the amount of each material used affects the outcome of the concrete compressive strength. In the database, there are a total of 1,030 rows for each of the columns. For all of the prediction columns, their respective minimums are 102, 0, 0, 121.75, 0, 801, 594, 1, and 2.331808. As for the respective maximums, they are 540, 359.4, 200.1, 247, 32.2, 1145, 992.6, 365, and 82.599225. In our solution to your problem of testing different combinations to compute different concrete compressive strength, we will code different data models to compute the best one for the mixtures that predict the best outcome. We will include two different interfaces that allow users to input the values of predictors to get a concrete compressive strength. The other interface will allow the users to include their observations into the database and automatically update the data model chosen to incorporate the new observations from the user. Below we have a list of our tasks and we go into detail about what each task accomplishes.

Database Exploration

Obtaining the database into coding for further use. (Kahla)

* Read the file and describe it, we can accomplish this by importing the pandas package as pd, and using pd.read\_excel() to read the database into our coding system, this allows us to manipulate the numbers for use in analyzing the data and so forth.
* Change the column names to shorter lower cases such as Cement -> c by using df.rename to make coding easier.

Create linear models

Create the models (Whole team)

* Using the statsmodels package in python, we will create linear regression models for each of the 7 predictors. After we get the linear models we will be able to access the model of best fit.

Use goodness of fit metrics (Audrey)

* MSE (Mean Squared Error) is a test that measures the average of the squares of the error, or basically the average squared distance between the estimated values and the actual values.
* RMSE (Root Mean Squared Error) is the square root of the MSE and converts the units back to the original units of the output variable and can be meaningful for the description and presentation.
* R^2 (or R2, R Square) test is used to indicate the goodness of fit of a set of predictions to the actual values. This measure is also called the coefficient of determination.
* Pearson's R correlation coefficient is the covariance of the two variables divided by the product of their standard deviations. Pearson’s R is the most popular type of correlation coefficient, because “[it] shows the linear relationship between two sets of data.”

Predict the effectiveness of the linear models (Mishael & Caelan)

* We’re going to define feature\_cols as all the columns (X) and use the test\_train\_split from sklearn.model\_selection to split the data with regard to test size and random state then predict the target variable (concrete strength also Y).

Perform Logistic Regression (Caelan)

* To define LogisticRegression, we will import LogisticRegression from sklearn.linear\_model, define logreg, fit the function, and do a line of code that makes the function predict the Y output from the X values.

Import the metrics class (Kahla)

* From the sklearn package we import the metrics class and define a cnf\_matrix, confusion matrix, and use the y predicted values and the y test values to produce a chart that shows the error types out of two classes.

Confusion Matrix & Classification Report (Audrey & Kahla)

* After defining the classes [0 & 1], use a heatmap function to help visualize the results then use the Y test outcomes and Y prediction outcomes with metrics to print the accuracy, precision, recall, and F1- Score which will evaluate the model. Lastly, import classification report and use Y test and Y predictors to get the grid.

Build user interfaces

Define the line function (Mishael)

* Once the linear regression lines from the models have been determined, we will create a function in python to define the line.

Interface for user input of concrete mixtures to predict concrete strength (Whole team)

* Build a code that allows users to input different predictors that generate output the concrete strength. Tools to help us would include for loops, if/else statements, and try/except structures.

Interface to update models for user observations (Whole team)

* Build a code that allows the users to input their observations that they have made on their own, and it will update the data model to include their observations with the underlying data that is already being used for the data model.
* **Citation(s)**:
* Glen, Stephanie. “Correlation Coefficient: Simple Definition, Formula, Easy Calculation Steps.” *Statistics How To*, 6 July 2020, www.statisticshowto.com/probability-and-statistics/correlation-coefficient-formula/.
* Unknown, Unknown. “Read Xls with Pandas.” *Pythonspot*, 29 Mar. 2017, pythonspot.com/read-xls-with-pandas/.