I decided to work to build a linear equation that can guide us to predict an output given multivariate data. My dataset of choice contains a consistent set of daily recorded data points over a number of weeks for attributes including temperature, wind speed, humidity, day of the week, presence of a holiday, and number of bicycles rented, which we will use as our output, given the presence and ‘weights’ or values of some of these factors. In an algebraic sense, I am solving for the output y, that being bikes rented, given the linear equation I will generate via a simple machine learning algorithm; Utilizing selected key attributes of the data set, who’s values per item will each be multiplied by coefficients that will be determined by the linear regression for the purpose of predictability. Given x, and z factors, what can we foresee for the result, y.

I built a linear regression with gradient descent in C++ for this purpose. The program takes a data file as input, builds an array and then splits out 75% for the training phase, and keeps the remaining 25% for the purposes of checking the standard error of the estimate, otherwise known as the accuracy of prediction. To build the linear formula the program I built splits the output column from the training data matrix, adds an identity row to the first of the data matrix (in a linear equation the first variable is essentially a constant) instantiates a vector of test coefficients to each begin at 100, along with a hypothesis matrix. The gradient descent at min cost function tests the actual output y in a given row against the hypothetically created y, and utilizes the cost function and each data point with a derivative based mathematical approach utilizing a constant in the equation, and a learning rate, to see what the ‘cost’ is given the hypothetical coefficients. I recursively run this gradient descent function, passing in the matrices and vectors as necessary, to build the training cycle for determining the coefficients for the linear prediction equation, until the cost function shows ‘convergence’, or that the tolerance of such is within a small range. I added into the algorithm a dynamic adjustment to the learning rate, as well. I then calculate the standard error of the estimate utilizing the found linear equation and the remaining data set. This too is a mathematical equation, albeit simpler in that it just compares the predicted values to the known values, and utilizes a tried and true equation to sum up the margins of error in each item.

To visually illustrate the correlation between attribute values per item, known output values, and the predicted linear regression per item, I utilized MATLAB. I wrote a script that plots items on the x axis and values on the y axis, showing for each item, it’s attribute values as ‘x’s in colors consistent for their respective parameter throught the set. I plot then the known output as a black circle for this item, and then depict the linear regression (based on the output of the program) as a line. This visually illustrates how the linear regression calculated programmatically fits the data.

I selected vector based matrices for my data structure for this project. These are matrices comprised of a vector holding vectors, i.e.. a list of lists. They provide the organized, quick read complexity of arrays at the order of O(1), while rising above the array structure in being able to provide flexibility in length, allowing us to add items without explicitly defining and redefining the size of each vector before adding data to it. Other data structures provide a larger order complexity in reading data, making algebraic calculations on them timely and data retrieval for them overly complex. In our mathematical equations in the linear regression we are accessing data based on index, based on position in rows and columns, and we don’t need to retrieve data based on searching for any number in particular. Data structures outside arrays shine in providing quick access to things we know we are looking for. Machine Learning involves modeling algorithms and performing calculations on all data in large sets. So long as we are careful in splitting our data sets into appropriate groups, we can successfully utilize the benefit of these structures having quick complexity in ‘reading’ data, and iterate through rows and columns of vectors succinctly within a couple of loops, utilizing consistent facts, such as all rows being items, to know that at index, ‘i’, a data item in one matrix, such as a data matrix would be found which pertains to the item, as well as that in a vector or matrix of a different category, such as output, give us data pertaining to the same item. This is useful since we can remove extraneous ‘for loops’ from our calculatations, and reduce the complexity of a program that way. We know these loops are based on the length of the data, n, so we know that they can make a program very costly in time resources.

I had a quick learning curve on working with vectors and vector operations on this project. Utilizing correct commands such as .push\_back, to update and add to vectors, and utilizing the iterator notation including vector\_name.begin() in associated vector building and updating functions was an interesting evolution. I learned that deleting rows or columns using .clear() can leave null values which break subsequent code, and notation based aspects including the importance of providing the necessary spaces in vector based matrices for the compiler. On a more conceptual level I learned a great deal about creating a training cycle based solely on data in machine learning, and the important leveraging that vector based arrays in C++ provide for this endeavor. I was a blank slate ready to absorb knowledge, and approached building my algorithm from a very specific design outline before I coded even a bit. This proved to be a very successful approach for me and it reinforced the importance of outlining and sketching out higher level architectural and design plans for a program before beginning. That process left me with a clearer, more definitive approach and an easier effort in debugging when a somewhat inevitable bug surfaced.

Specific challenges I encountered along the way with the algorithm itself that left me with very late night, and sometimes early morning edits were challenges with more multivariate data. I have come to the conclusion that although vector based matrices offer faster processing speeds over other possible data structures, they still utilize significant processing resources to run. With more multivariate data, the algorithm will need to ‘train’ iself through an exponentially higher level of cycles to find a solution, and although it may come upon what we could see as a ‘correct’ solution mathematically through the process (i.e., providing a low result in a cost assessment), it may not find the solution that makes the most sense to us or that provides the lowest error on initial run. There are factors including the ‘learning rate’ and the starting point which effect the response, and although I build the learning rate to have a manual switch that when added to the function as false, would cause the learning rate to increase by .0001 after a set number of cycles of the algorithm, or to decrease if the coefficients began to drop into a reverse range, I think that defining that aspect of the program at a higher level with a more sophisticated approach could provide substantial benefits in finding not just the best mathematical fit given a certain learning rate and starting point, but a more accurate fit in terms of what would make the most sense given the data set, in a more timely fashion. I also learned that although the beauty of working with the vector based matrices is in the fact that we get to utilize a minimal number of loops constrained to a row and then a column within each row, that this aspect can become confusing when you have multiple sets of data. Further designing a visual reference for these sets before coding might make sense, to reduce initial bugs of which the root is tricky to deduce, when you are trying to visualize the rows and columns of the multiple vectors and matrices involved. It would require some legwork, however especially in more complicated problems this would prove to be a lucrative endeavor in that it would create efficiency in initial accuracy of the program. I think the awesome thing about the structures I selected is that they are uniform, quick, easy to organize code around, and easy to apply mathematics to data on utilizing multiple sets, however the drawback is that everything isn’t labeled easily with a name or a easy to reference label, so one can get a bit lost in the rows and columns. One feature that would be nice to build would be a naming reference array to define that a row in one matrix might identify an item in the data set, where as a column might have a certain list of names. And to keep this list based on the same order. I think the ordering is easy to keep uniform throughout a program, however having a reference programmatically as well as visually for the developer would prove to be very useful and provide a level of orientation within what becomes a program with multiple levels utilizing data as well as mathematics. Keeping the approach specific and the labels accessible while preserving the flexibility and quick offerings of utilizing the simplicity of vector based matrices via reference to data in terms of row and column index, would allow for balance and more confident development as well as more minimal bug finding as a program becomes more complex.

I learned the importance of thorough research, defined and a strong design approach as well as a clearly defined implementation approach from this project. I was careful to really understand the problem I was facing before I began to code, and to outline a structure to build from, first. With a new topic to explore and a great deal to learn syntaxically from working with a new data structure, putting in the leg work for thorough initial research proved to make this project possible for me, and I learned that the efficiency I gained from this enamors me to this process, greatly. I sketched out not only on paper but in code, my approach and the equations I would be using. I notated columns and rows clearly with a consistent notation of ‘i’s and ‘j’s. I have literally started following suite to this process in a work project which was made complex by what I think was an very inappropriate and out of touch design decision by another developer, causing a more cohesive adaptation in approach to be necessary, and thereby making this outlining approach although seemingly neurotic at first, extremely useful for reference and for keeping multiple elements in sync and organized in the context of a deliberately defined greater plan. I also have learned the importance of labeling and allowing for further flexibility and evolution in a program. I worked myself into a certain structure utilizing my recursion which made me start thinking about how I would like to be able to utilize references to structures better, if I was not in fact returning an item always from a function. I think I could learn further on improved design patterns for situations where one would like to both perform business logic on a set of structures as well as return data from a function and combinding this or separating functions to allow for optimal setup for this sort of thing. In the meantime I have begun to think further about data storage and creating references to data to alter their structure, which has been helping me further understand how I will continue to improve my understanding of utilizing ngRx in Angular on the front end in my job, currently, to transform data when appropriate in the ui state as well as the data state to better serve the interactivity and UI of the app, as well as to best centralize the data while not overarchitecting. That is an interesting balance for me, and one that I am still contemplating and exploring. The main takeaweay being that in some instances definitiely creating an object ‘by reference’ or storing it in the ui state, although timely and requiring further organization, can become a great strength to an application that contains further complexity within a controller, and which can abstract out a level into simplicity that can otherwise make code writing laborious if attempting to keep the variable rebuilt and rescoped by name in every localized transformation.

An important takeaway from this project for me has also been learning the importance of using an appropriate data structure for performing repetitive and converging calculations within and amongst multiple sets. I am really amazed at the extreme ease the vector based matrix structure makes this endeavor almost mindless and with careful structural setup it can be absolutely amazing in implementation.