**Using a Machine Learning Model (Linear Regression) to Predict Days to Neutrophil Recovery After Blood or Bone Marrow Transplant in Pediatric Patients**

Computer Science Capstone

July 17, 2024

**Contents**

[Task A – Letter of Transmittal 4](#_Toc172159578)

[Task A – Project Proposal 6](#_Toc172159579)

[Task B – Executive Summary for the Information Technology Team 18](#_Toc172159580)

[Task C – The Application 25](#_Toc172159581)

[Task D – Documentation 27](#_Toc172159582)

[Quick Start Guide 57](#_Toc172159583)

[References List 60](#_Toc172159584)

# Task A – Letter of Transmittal

WGU Hospital

300 Mountain Road

Los Angeles, CA 90035

Dear Dr. Smith,

I am currently a data analyst working in your hospital within the Blood and Bone Marrow Transplant (BMT) Program. I am composing this letter to you, our chief medical officer, to propose the implementation in our hospital’s workflow of a data product that I have designed.

As we both know, WGU Hospital is renowned for its blood and bone marrow transplant program. This stem cell transplant program benefits people with hematologic and/or malignant disorders whose bone marrow or immune systems are damaged and who usually have not responded to chemotherapy or other conventional treatments. Therefore, improving our transplant program in any way would mean improving outcomes for these patients who have come to WGU’s specialized transplant program.

My proposal is to use a machine learning model to supplement the great work that is already being done. The product that I have designed uses linear regression to predict how fast a pediatric patient will recover cell counts after a stem cell transplant. The model would consider unique patient features to make its prediction.

The predicted value could then be used by physicians to help plan or anticipate how to modify treatment course after transplant. Knowing this value adds to a patient’s overall clinical picture and could aid physicians to more easily flag certain patterns or risk factors, which might improve patient outcomes.

In addition to improving patient care, a machine-learning tool such as mine would also benefit the hospital as well. The predicted number of days to cell count recovery can serve as a marker to anticipate or modify a patient’s treatment course more efficiently and effectively, which could mean better use of hospital resources to treat BMT patients (who often remain in the hospital for multiple weeks after transplant), thereby offering the possibility of financial benefit to the hospital as well.

Overall, good clinical outcomes translate to good performance indicators for our BMT program. Since our hospital’s performance gets published by accreditation bodies, any way we can improve our outcomes could potentially attract more patients and physician referrals to our hospital when they see how our BMT program ranks against other programs. Using every tool at our disposal to contribute to successful responses to transplants is essential for the ongoing support and accreditation of our BMT program so that we may continue to offer lifesaving treatments.

Please find attached my project proposal, to review at your convenience. I appreciate your time and consideration and look forward to hearing back from you soon.

Sincerely,

[*Name omitted* *for privacy*]

# Task A – Project Proposal

Summary of the Problem

Blood or bone marrow transplants (BMTs) are stem cell transplants that are used to treat people with malignant or nonmalignant diseases of the blood or bone marrow. A BMT is usually preceded by a conditioning phase that intentionally wipes out both cancer cells and the immune system of the patient, including the blood-producing cells in the bone marrow. This allows a “blank slate” on which the patient can re-build a new and hopefully improved immune system from the transplanted cells. BMTs can be autologous (come from the patient) or allogeneic (from a donor). In either case, the stem cell transplant, also called a graft, needs to engraft (i.e., the stem cells travel to the bone marrow, divide, and make new blood cells). As engraftment occurs, cell counts start to recover.

One of those counts is (automated) neutrophil count, often seen as ANC on lab reports. Ideally, ANC keeps increasing and recovering after transplant until it goes above a certain threshold. Then, the patient is considered to have ‘ANC recovery’ and is considered ‘engrafted’ which is a major goal after transplant.

ANC recovery and the upward trend of ANC after transplant is a major indication that the graft is working well and re-building the immune system as planned. Engraftment typically happens around 15 to 30 days after transplant, but in some cases engraftment can be delayed due to many other factors, such as infection, medicines, or failure of the graft (Stanford Medicine, n.d.). Please note that although engraftment usually happens in those first few weeks after transplant, actual full recovery of the immune system can take months to years.

The problem I am trying to solve with my data product is not so much a problem as it is an opportunity to enhance WGU Hospital’s current BMT program. Clinicians know BMT patients should typically recover ANC around day 15 to day 30 after transplant; this is commonly known and expected. However, if we could more precisely predict when a patient with a particular profile of personal characteristics might recover ANC after transplant (for instance, being able to predict if ANC recovery will be closer to 15 days versus closer to 30 and beyond), clinicians would have one more data point to add to a clinical picture and one more tool to enhance their clinical decision-making. Since a lot of BMT treatments are new technologies or clinical trials, there is not as much data for clinicians to draw guidance from as compared to the wealth of data available on conventional treatments. Perhaps, a machine learning product to help provide some insights based on historical data and thus bridge this gap would be useful to our BMT physicians.

Benefits to the Customer

The BMT physicians, the hospital, and the BMT patients could be considered as the customers.

For the physicians, as discussed above, there is value in a machine learning tool that can pinpoint when a patient might engraft. It might assist the physicians in planning patient treatment after transplant, or to more easily anticipate modifying treatment course if a patient is not recovering ANC when predicted.

For example, if the patient’s predicted ANC recovery day of Day 13 has come and gone and there is no sign of ANC recovery, it could be a flag for the clinician to consider changing treatment course a few days earlier than if they had waited for the general 15 to 30 days. Possibly, this might lead to initiating necessary medication changes or treatments earlier rather than when it is too late. Even shortening inpatient length of stay by one day could be beneficial if we consider that BMT patients typically recover from transplant in the hospital, translating to weeks of continuous medical care that can be costly for patient and hospital alike.

As another example, if the physician knows a patient is likely to have ANC recovery at day 13, then certain procedures, tests, medications, or schedules can be managed in anticipation of that. Thus, knowing more accurately when the patient might engraft could potentially help with planning and scheduling, which in turn helps to use hospital resources efficiently. BMT patients not only have extended stays in-hospital but also have recurrent medical needs and followups once discharged, so they will be using medical services possibly for years. If my tool provides a way to reduce any of those costs by using resources more efficiently, then it will benefit WGU hospital as well as the patients.

Outline of the Data Product

As a general outline, the data product that I have designed uses a machine learning model to predict days to engraftment (measured by days to ANC recovery) in pediatric patients. The program uses a linear regression model (multiple regression) and uses a public dataset provided by the University of California, Irvine (UCI) for training and testing the model. The program’s basic operation is that it takes in a set of patient characteristics and outputs a predicted number of days to ANC recovery based on that profile. The program will be written in Python within a Jupyter Notebook, which will be shared in Google Colab.

Description of the Data

I will now describe the data that will be used in the making of my data product. University of California Irvine has various publicly available machine learning datasets. The UCI dataset I will use to build my data product is called “Bone marrow transplant: children.” This de-identified dataset contains information about a set of pediatric BMT patients with hematologic disorders, both malignant and nonmalignant. All patients received allogeneic unrelated donor stem cell transplants.

The downloadable dataset provided by UCI is a ‘bone-marrow.arff’ file. UCI also provides the option to import the dataset directly into your code, which is the method I used. Once imported into my code as a dataframe by pandas library, I exported it to my local machine as an Excel file for a quick manual viewing so that I could get an initial sense of the whole dataset; this was possible for this project because there were only 187 records, or patients, in the dataset.

A table of information

Description automatically generated

There are 37 columns or variables, which are viewable in the Jupyter Notebook screenshot above. Generally, these variables are patient information, donor information, disease type, and measures for responses to transplant. Based on my domain knowledge, consulting with my manager who is a subject matter expert, and my own research online, I have narrowed down the independent variables to be used in my machine learning model to: source of stem cell, dose of cells transplanted, blood type match, certain genetic/immunologic matches, and diagnosis being treated. As described previously, the dependent variable I will predict is the number of days to ANC recovery, which is the timepoint when neutrophil count recovers to above threshold of 0.5 x 10^9/L.

While the dependent y-variable ANC recovery is continuous, most of the independent variables are categorical. I will have to encode these to numerical values so that they can be used by my machine learning model.

Objectives and Hypotheses

I will be using a linear regression model to predict the dependent variable of ANC recovery days. Since there are many independent variables predicting the ANC recovery, this is called multiple regression. The regression equation model is below; without getting too detailed, the B’s are coefficients for each predictor variable (the x’s) and B0 is the y-intercept.



In multiple regression, the null hypothesis H0 and alternate hypothesis H1 mathematically look like below:



As seen, the null hypothesis is that all coefficients are zero, meaning that none of the variables “belong” in the model; the null hypothesis claims that there is no significant relationship. The alternate hypothesis is that at least one of the variables belongs in the model, meaning there is at least one significant relationship of an independent variable to the dependent variable (Jones, 2004). My hypotheses are described in the table below.

|  |  |  |
| --- | --- | --- |
| **VARIABLE NAME** | **NULL HYPOTHESIS H0** | **ALTERNATE HYPOTHESIS H1** |
| **Main hypotheses** | **None of the dependent variables (see below) have a significant effect on ANC recovery** | **At least one of the dependent variables has a significant effect on ANC recovery** |
| “Stemcellsource” | No significant effect on days to ANC recovery between stem cell sources | Stem cell source has a significant effect on ANC recovery days |
| “CD34kgx10d6” | Cell dosage has no significant effect on ANC recovery days | Cell dosage has a significant effect on ANC recovery days |
| “ABOmatch” | Blood type compatibility has no significant effect on ANC recovery days | Blood type compatibility has a significant effect on ANC recovery days |
| “Antigen” | Number of antigen mismatches has no significant effect on ANC recovery days | Number of antigen mismatches has a significant effect on ANC recovery days |
| “Allele” | Number of allele mismatches has no significant effect on ANC recovery days | Number of allele mismatches has a significant effect on ANC recovery days |
| “Disease” | Disease diagnosis has no significant effect on ANC recovery days | Disease diagnosis has a significant effect on ANC recovery days |

My objective for my data product is to accurately predict ANC recovery days for a unique pediatric patient profile. However, I do expect that the accuracy of my model will be low due to the small dataset of 187 records. I anticipate having to drop several rows due to null values, which will make the dataset even smaller. Small datasets can have bias that larger and more diverse datasets do not have. Therefore, I do have concerns about what the accuracy of my model will be.

Project Methodology Outline

While many project methodologies exist, such as CRISP-DM and Agile, my data product is for use within my own team and home department. Also, this will be a small project, with the main collaborators being the software engineer who will code it, the data analyst (myself) who will provide the data insight and analysis, and likely a couple other members from my data team to help with testing the software. Due to the informal and small setting of our department, even the development of a department-wide project such as mine would be best served by a methodology that is simple, straightforward, and less formal.

Therefore, for this project, I will be following the steps of the Data Science Lifecycle (Institute of Data, 2024), which are: (1) defining the problem, (2) data collection and preparation, (3) data exploration and analysis, (4) model building and evaluation, and (5) deployment and maintenance.

For step 1, the main problem is that there is an opportunity to integrate a machine learning tool into BMT physician workflow. The tool would aim to better predict days to ANC recovery after pediatric BMT.

For step 2 of the data science lifecycle (data collection and preparation), I will import UCI’s public pediatric BMT data into my data product. Pre-processing the data would be very important. Cleaning the data and also reducing dimensionality or features will be important. This step will benefit from subject matter expert insight, which would mean using some of my domain knowledge and online research but also consulting with my manager or our BMT physicians to determine if there are relevant clinical relationships between the variables. I will also have to determine how to handle empty or null values based on the nature of the variable; I will draw on my own knowledge, but I recognize that, again, this task would benefit from consultation with clinical experts.

Step 3 is data exploration and analysis. I will do some exploratory data analysis to see what kind of relationships exist between variables. I will calculate some basic statistics but also create some graphs and visualizations to get a better idea of any relationships that exist between variables.

Step 4 is to build and train the linear regression model by splitting data into train and test sets, and then evaluating how well the model works. I will be using Python machine learning libraries (scikit-learn) for the data split and fitting the model. For evaluation of the model performance, linear regression models cannot be analyzed for accuracy in the same way as classification problems. Rather, instead of measuring accuracy like we do for classification models, we calculate error, such as Mean Absolute Error and Root Mean Squared Error (Brownlee, 2021). In linear regression model evaluation, it will be important to analyze residuals as well. Residuals are the difference between the actual and predicted values (Ogunbiyi, 2022).

The last step of the data science lifecycle is deployment and maintenance. I estimate six months from start of the product development to product deployment. This step will be dependent on hospital leadership’s approval to integrate this tool into BMT physician workflow and how strongly they want to continue to put resources toward the use of this product after deployment.

Funding Requirements

(*Note to WGU evaluator: I based my cost estimates on my real-life work environment in a mid-to-large size BMT program that uses Google Cloud Platform as its cloud service.*)

WGU hospital already has equipment in-place for the normal workflow of its BMT staff (desktops, monitors, laptops, and cloud services for data storage via Google Cloud Platform), so expected costs of this data product are largely labor costs and any extra on-demand costs for using Google Cloud Platform’s cloud GPUs or tools.

Using GCP’s price estimator, 100 hours of GPU use per month per 0.14 GPU will cost about $70. If my data product were to be deployed and integrated into the hospital workflow, I expect hours of use to be far less, as I imagine every time the product is run for a prediction and uses cloud GPU, it will be under one minute. Our BMT program has about 30 physicians and about 50 BMT patients a month. If the physician group collectively run this program once daily for 50 patients a month, that is still under 1 hour of GPU usage per month. Additionally, my product is designed for pediatrics, which is a subset of the larger BMT program, so it will have much less usage than the whole BMT program. To err on the side of grossly overestimating usage than underestimating usage, I will predict GPU costs to be about $70 per month, or $420 per half-year.

We will likely need to hire a software engineer for a contract of six months to develop the program and user interface and to integrate the tool into existing hospital software. The yearly salary in California for a software engineer is about $180,000 according to Glassdoor. If the project takes half a year to develop and deploy, then expected cost for the first six months is $180,000/2 + $420, or $90,420 total on top of the budget the BMT program has already.

The machine learning libraries and software used to develop this data product are free and will not impose a cost. Examples are Jupyter Notebook, scikit-learn, pandas, matplotlib, seaborn, numpy, etcetera.

Once deployed, I expect maintenance costs to continue to be labor costs and use of cloud GPU. The software engineer would not need to be full time once the project is deployed, so they may be reduced to part time hours to maintain and troubleshoot the project as needed. Therefore, expected maintenance costs per year will essentially reduce to the number of hours the software engineer works.

The Impact on Stakeholders

The stakeholders are the BMT physicians, WGU Hospital, and the BMT patients. If this data product is approved and deployed, BMT physicians will need to be trained on its use. I recognize there is a chance the physicians might not use it. From my experience, different attempts to change physician workflow have failed due to poor physician adoption. However, I envision a very easy interface will be integrated into the BMT database console that our physicians already use, which means it would be very easy for the physicians to integrate my tool into their workflow.

Once the physicians start using the data product, the secondary impact will be on the hospital and the patients. As discussed previously, if use of this tool improves clinical outcomes and reduces medical costs and use of hospital resources, then it will have a generally positive impact on all stakeholders involved.

Ethical and Legal Considerations and Precautions

The UCI dataset is a publicly available dataset with all protected health information removed. As such, there is no precaution around using the data in terms of patient identifiers.

Another consideration, however, is the impact a tool like this could have on patient’s lives. For instance, consider if physicians modify the course of a patient’s treatment based on the tool’s prediction, but the treatment modification leads to a negative patient outcome. In such a case, does the risk outweigh the good? I caution the use of this data product over sound clinical judgment, as it is simply a tool and should not replace clinical skills and reasoning.

My Expertise

(*Note to WGU evaluator: This part is true; I do currently work at Stanford University’s BMT and Cell Therapy Program as a data analyst, and I did consult with my manager for extra insight on my topic.)*

Prior to getting hired as a data analyst for WGU Hospital’s BMT program, I worked in Stanford Medicine’s BMT department for two years. I will rely on my domain knowledge of BMT to help me with data cleaning and deriving meaning from my results. I also consulted with my manager to obtain more insight into the topic of BMT and help inform the design of this project.

In addition, through my studies, I have knowledge of coding, machine learning concepts, and how to apply them to data, so these skills will help to develop and design the code for my data product.

# Task B – Executive Summary for the Information Technology Team

The Problem or Opportunity to be Solved

Neutrophil count (ANC) is one of the major indicators to track how well (or not) a patient is recovering after their blood or bone marrow transplant (BMT). The subject of this project proposal is the development and integration into current hospital workflow of a machine learning tool that predicts the number of ANC recovery days for pediatric BMT patients.

Customers and Customer Needs

The primary “customers” for this data product will be the BMT physicians. The physicians will benefit from a tool in their clinical toolbox that will more precisely predict days to ANC recovery than the common knowledge that 15-30 days is generally expected for BMT patients. Being able to better anticipate when a patient will engraft will help physicians to efficiently plan or modify treatment course after the transplant.

Gaps in Existing Data Products

Currently, our WGU Hospital BMT program does not use any machine learning tools. As far as existing data products, we do have a database of patients and their transplants. In the future, if approved by hospital leadership, this wealth of in-house data could potentially be used as a historical data source for my new data product. But, for the purposes of this project proposal, we will initially build and train our machine learning model with the UCI pediatric dataset.

Also, WGU Hospital’s BMT database console does have a few analysis features, including some simple pre-built graphs and options to customize certain automated visualizations. I envision my data product to integrate into this console. This way, it would be easy for the BMT physicians to incorporate the new tool into their workflow.

Data Availability or Collection

For this data product, a publicly available pediatric BMT dataset from University of California Irvine will be used for initial development.

Methodology to Guide Data Product Design and Development

For this project, I will be following the steps of the Data Science Lifecycle (Institute of Data, 2024), which are: (1) defining the problem, (2) data collection and preparation, (3) data exploration and analysis, (4) model building and evaluation, and (5) deployment and maintenance.

For step 1, as described previously in Task A’s Summary of the Problem, the main problem I am addressing is a gap in WGU Hospital’s BMT program workflow for a tool that can better predict days to ANC recovery after blood or bone marrow transplant in pediatric patients.

For step 2 of the data science lifecycle (data collection and preparation), I will follow UCI’s instructions to import their module into my Jupyter Notebook with “from ucimlrepo import fetch\_ucirepo” and “bone\_marrow\_transplant\_children = fetch\_ucirepo(id=565)”. This will allow me make a dataframe from their dataset, which allows me to examine the data within Jupyter Notebook or export to .csv or .xlxs for review within my spreadsheet program. To clean the data, first, I want to take care of dimensionality reduction/feature reduction, as 37 variables (including independent variable ‘ANCrecovery’) is a lot of variables. Due to the specialized and very complex nature of blood and bone marrow transplantation, I will mostly rely on my domain knowledge, consulting with my manager who is a subject matter expert, and my own online research in order to decide which variables should make up a patient’s profile of characteristics to predict ANC recovery. I would then use Jupyter Notebook and Python to handle any missing values as appropriate (e.g., replace with zero, delete row, or replace empty values based on a mean, median, or mode, based on the nature of the variable).

Step 3 is data exploration and analysis. I will do some exploratory data analysis to see what kind of relationships exist between variables. This would include looking at data types, counts of rows and columns, some basic statistics like mean, median, range, etcetera. I would also want visualizations of the dataset to better see possible relationships among the variables. These would be graphs such as histograms, scatter plots, and pair plot (which would be a collection of scatter plots).

Step 4 is to build and train the linear regression model by splitting data into train and test sets (I will do an 80-20 train-test split of my dataset), and then evaluating how well the model works. Various statistics and visualizations can help with this part. For instance, I would do calculation of p values to get an idea of statistical significance. Additionally, I will include a scatter plot of the test values (actual values) versus the predicted values to see how well they match; a slope of 1 would indicate perfectly matched predicted and actual values. For linear regression, it is very important to analyze residuals; I will do this by plotting residuals and also calculating error metrics like Mean Squared Error, Root Mean Squared Error, Mean Absolute Error, and R-Squared (Ogunbiyi, 2022).

The last step is deployment and maintenance. This part will involve the continued employment of a software engineer after product deployment, perhaps at part time hours. Then, he or she could maintain or update the software as needed.

Deliverables

The first deliverable for this project will be a preprocessed dataset that is ready for use by a machine learning model. The next deliverable will be a trained machine learning model; this deliverable will be provided by the data analyst (myself) to the software engineer. The software engineer will next develop an easy-to-use user interface and the completed software application. During this time, the software engineer will collaborate with the data team (key members of the data team and myself) to test the application before deployment.

Plan for Implementation and Anticipated Outcomes

The plan for implementation of my data product will involve the software engineer developing the software that will incorporate my linear regression model. The plan is to integrate this software application into WGU Hospital’s existing BMT database console. My vision is for this machine learning tool to be available in a section of our BMT console which is easily accessible to the BMT physicians. I also envision a simple user interface so that non-technical users can easily use and understand the tool without technical support.

My expected outcome is that we will successfully integrate my tool into the BMT database console because it will likely be just another tab within the console. Another expected outcome is that the majority of physicians will use the data product because it is easy to use. Even if the physicians end up not relying heavily on the output of the program, I believe the number of days that the program outputs will be of interest regardless, just out of curiosity about ‘AI’ or the novelty of a new tool. I anticipate that in the beginning, this tool will not be accurate; the training data is far too small. My hope is that after this initial data product is rolled out for pediatrics, we will be allowed to use our in-house data as training data for the machine learning model. This would improve our model’s accuracy because our in-house database is much more extensive (10,000+ BMT patients versus the 187 in the UCI dataset).

Methods to Verify Requirements and Needs are Met

I determined in Task A that the business requirement or need is to more precisely predict when a particular profile of patient characteristics might recover their ANC levels.

I will verify that my data product meets customer needs by meeting monthly with the physicians to get their feedback on the tool. Since the software engineer will be available at least part time after deployment (as requested in this proposal), any issues to fix or update can be brought to his or her attention. If the physicians feel the tool is very inaccurate, I will need to review and revise my model.

Also, I will continue to add to the dataset as we perform more pediatric BMTs. Re-training the model periodically on more and more data will help the model to become more accurate over time because we would have more data to train on. This would mean I need to regularly calculate performance metrics such as R-squared (or adjusted R-squared), which essentially describes accuracy of the model (Jones, 2004), to see if the model is becoming more accurate.

Also, we have proposed a budget for this project. As a way to check if our project met this specific business requirement, I can do a cost-benefit analysis by calculating expenses and comparing them to any financial (or other) gains possibly made by use of this tool. I could compare if our survival numbers or patient in-hospital days have improved since integrating my data product into the workflow. I could additionally compare if we received more patient referrals or performed more transplants per month than prior to the tool because these would be a sign that our program has good outcomes and is attracting more people. It is difficult to measure the impact of my tool because it only outputs one value, which is one datapoint within a very complex medical system, but any of the items I mentioned might be indirect indicators that my tool has utility.

Programming Environment and Related Costs

The programming environment will be Python and Jupyter Notebook to code the machine learning model and build the data product. Python and Jupyter Notebook are both free, and free machine learning libraries such as scikit-learn will be used as well. The plan is to integrate our data product’s interface with our existing BMT software. We will use Google Cloud Platform to provide the compute for calculating ANC recovery days via a machine learning model.

Since the goal is to integrate this product into our existing BMT workflow, we already have equipment such as computers and a BMT database software application. My plan is just to add our data product onto this framework. Therefore, costs will be limited to labor and on-demand cloud usage.

I expect the project to take six months to complete. Costs will include pay for a six-month contract for a software engineer (about $90,000) and the on-demand cloud GPU usage (maximum $420 per six months).

Projected Timeline

Please find below a projected timeline for the first year of this project.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Start Date** | **End Date** | **Duration** | **Dependencies** | **Resources** |
| Discuss project goals with physicians and other involved staff (nurse coordinators, BMT data team, hospital administrators, etcetera) to get input and any constructive feedback. | Jan 1, 2025 | Feb 28, 2025 | 2 months | The schedule availability of key staff members | 3-6 meetings with all involved staff to inform of project goals and initial progress and get feedback on design |
| Milestone: Collect data, clean data, exploratory data analysis | Jan 1, 2025 | Jan 31, 2025 | 0.5-1 month | We will use the public pediatric dataset as a starting point. However, if approved by hospital administration, we could also use our in-house dataset of thousands of BMT patients from the beginning of our BMT program. | BMT Data analyst to collaborate and communicate results to software engineer |
| Milestone: Design, develop, train, and test accuracy of machine learning model | Feb 1, 2025 | Feb 15, 2025 | 0.5 month | Dependent on quality data preprocessing | Data analyst, Software engineer |
| Milestone: Complete development of the software and user interface. Integration into existing BMT database console. | Feb 15, 2025 | May 31, 2025 | 3-4 months | Regular testing of the new software by select users to give feedback to software engineer for next iteration of product improvement before deployment | Software engineer. Members of existing BMT data team (data team leadership and BMT data analyst) to act as users. |
| Milestone: Deployment of program | June 1, 2025 | June 30, 2025 | 1 month (to include trouble-shooting issues after deployment) | Go-live date dependent on hospital administration and BMT physicians’ approvals | Software engineer to troubleshoot any issues after deployment |
| Maintenance | July 1, 2025 | Onwards | 6 months remain in year | Dependent on any issues that come up or requests for certain features by users | Software engineer |

# Task C – The Application

Please find below an overview of application requirements and how I met them. These points are found within the Jupyter Notebook itself; please refer to the Jupyter Notebook for more details.

|  |  |
| --- | --- |
| **Requirement** | **My Application** |
| • one descriptive method and one nondescriptive (predictive or prescriptive) method | * Descriptive method – basic statistics (mean, median, etc.), bar plot (e.g., Section B graph called ‘Average ANCrecovery by Disease Category’, Section C graph that shows distribution of ANCrecovery), pairplot of variables * Predictive method—Random Forest for feature importance, Linear Regression Model for predicting ANCrecovery |
| • collected or available datasets | * Public dataset of pediatric BMT patient data available from University of California Irvine |
| • decision support functionality | * By predicting a number of days to ANC recovery, physicians have a metric that they can use to anticipate how to prepare for the predicted date or adjust treatment course due to the patient meeting or not meeting the predicted date. |
| • ability to support featurizing, parsing, cleaning, and wrangling datasets | * My Jupyter Notebook describes the steps I took to reduce number of features and how I cleaned the data. |
| • methods and algorithms supporting data exploration and preparation | * I used domain knowledge, consultation with a BMT subject matter expert, and Random Forest algorithm to assist with reducing number of features. * I used various methods to explore the data, including histogram and pairplot. |
| • data visualization functionalities for data exploration and inspection | * I used barplots, histogram, and pairplot. |
| • implementation of interactive queries | * Throughout my notebook, you will see visualizations or “print outs” of output at nearly every step. |
| • implementation of machine-learning methods and algorithms | * Linear regression was my main machine learning model * I also used random forest algorithm to help with feature selection * I also used ordinary least squares which is a method for finding the coefficients of a linear regression |
| • functionalities to evaluate the accuracy of the data product | * Ordinary Least Squares for p-values and R-squared * Calculations of R-squared, Mean Absolute Error, Mean Squared Error, Root Mean Squared Error |
| • industry-appropriate security features | * All patient information is de-identified. * I will be distributing a Jupyter Notebook via Google Colab to WGU evaluators only. I will rely on Google Colab’s ability to restrict access to my file based on my permission settings. |
| • tools to monitor and maintain the product | * A way to monitor the product in the future would be to keep testing my model, perhaps changing the random\_state number of my model to get a different ‘randomization’ of data and re-calculating all error metrics to re-evaluate my model’s performance. Additionally, as more pediatric BMT data is obtained, adding it to the dataset and re-training my model would be helpful and possibly improve accuracy. Similarly, further maintenance might involve obtaining even more data (getting approved for access to bigger datasets). * As for within the product itself, I have added various calculations and methods to evaluate my model’s performance. They can be accessed by running the particular cells within the notebook to see the output of the calculation/metric or graph. |
| • a user-friendly, functional dashboard that includes three visualization types | * I have implemented ipywidgets to provide a simple interface for user input with mostly dropdown menus and one text box. |

# Task D – Documentation

Business Requirements Document

I have provided a business requirements document to illustrate the business aspect of my data product; there are 7 key components that I have outlined below (Asana, 2024).

|  |  |
| --- | --- |
| **Executive Summary** | My data product aims to provide clinicians with a machine-learning tool that uses a profile of patient characteristics to predict days to neutrophil recovery in pediatric blood and bone marrow transplant patients. |
| **Project Objectives** | Predict days to ANC recovery within 3 days of error (Mean Absolute Error).  Decrease typical inpatient length of stay after transplant by one day. |
| **Project Scope** | The BMT data team will discuss with the BMT physicians and software engineer about project design and goals. The data analyst will preprocess the dataset and build and train the machine learning model in collaboration with the software engineer. The software engineer will develop and deploy the application, which will be integrated into existing BMT software. I expect the project to be complete and ready-to-deploy in six months, with costs not to exceed $90,420 in the first six months. |
| **Business Requirements** | * Preprocess dataset – Priority level 1, Critical level High. * Build and test machine learning model—Priority level 1, Critical level High. * Integrate software into existing BMT software, with user interface—Priority level 1, Critical level High * Discuss initial project goals and design with key staff—Priority level 2, Critical level Medium. * Application maintenance—Priority level 3, Critical level Low. |
| **Key Stakeholders** | * BMT physicians who will be the main users of the data product * Hospital administrators who will provide project approval and keep an eye on ongoing quality review * BMT patients whose treatment outcomes may be impacted by use of the data product |
| **Project Constraints** | * Timeline: Six months to deployment * Budget: Not to exceed $90,420 for first six months. Maintenance costs after deployment will be less due to software engineer will go to part-time hours. * Team Availability: Must follow software engineer and BMT data team’s scheduled work hours * Project Risks: If the machine learning model has poor accuracy based on the available data, utility of the data product is decreased |
| **Cost-benefit Analysis** | Cost || Benefit   * Team member work hours || Team members develop a new tool that will be integrated into existing hospital software, which will create impact for years to come and improve patient outcomes (and our scores on audits for BMT program accreditation). * Committing to use of cloud services || Using Google Cloud Platform’s compute and storage means less costs towards onsite equipment   If use of the data product increases efficient use of hospital resources, which might translate to fewer inpatient days by the patient after transplant. Also, if use of the data product improves our standing against other BMT programs, then patients and referring physicians might seek out our program over others. Even if we only get 1 more BMT patient per month who chooses our program, that would mean an extra insurance reimbursement. Current and exact numbers are difficult to find, but according to a letter written by AABB et al. in 2018, there was about $20,000 left to provide 30 days of care for each Medicare BMT patient. The letter illustrates how costly BMTs can be for a hospital, sometimes meaning the hospital goes into “the red” to care for their BMT patients. It follows that a hospital would aim to decrease BMT costs wherever possible.    **Total Cost** = $90,420/6 months, or $15,070/month  **Expected ROI** = To be determined, but possibly $20,000/month if we increase patient referrals by one per month  (However, the $20,000 is just a ballpark figure before expenses and can vary incredibly widely, depending on how costly the patient’s medical care is, which is a difficult number to assess when these patients have a lot of medical needs and often get followup care for anywhere from months to years to lifelong.) |

Raw Dataset

The original and raw dataset provided by University of California Irvine (UCI) can be downloaded from this link (<https://archive.ics.uci.edu/static/public/565/bone+marrow+transplant+children.zip>) or this link (<https://archive.ics.uci.edu/static/public/565/data.csv> ). The zip file contains a .arff file with metadata. UCI also provided code to import the dataset directly into my notebook. Once I did, I was able to read the metadata of the dataset. In the metadata there is reference to a url to a csv file. Using pandas, I made a dataframe from the csv at this url.

A white background with black and green text

Description automatically generated A close up of a computer screen

Description automatically generated A screenshot of a computer

Description automatically generated

Cleaned Dataset

A major part of data-preprocessing (getting the data in a form ready for analysis or use by a machine learning model) is cleaning and transforming data. My objectives to clean and transform the data were to (1) reduce number of variables, (2) handle missing values, (3) handle values that didn’t make sense or may be incorrect, and (4) encode categorical variables.

Dimensionality reduction or feature selection is very important to minimize how complex data is in order to increase model performance. So, dimensionality reduction methods reduce the complexity of the data. One way to do this is feature selection (though I note that varying sources sometimes describe feature selection as being distinct from dimensionality reduction). Feature selection involves reducing the number of variables in your dataset in order to improve interpretability of the machine learning model (Sanjyal, 2022).

Based on my domain knowledge, consultation with my manager who is a subject matter expert, and my own online research, I determined six independent variables (see below) to be most relevant in predicting the dependent variable “ANCrecovery.” 

Next, I wanted to handle missing values. I wrote code to tell me exactly which columns I should examine and how many rows were involved; the output tells me that the columns ['Antigen', 'Allele', 'ABOmatch'] have missing values. Blood type match and the differences in donor match for antigen and allele should not be estimated or guessed, so I decided to drop the rows with missing values. After dropping rows, my record count went from 187 records to 180.

A screenshot of a computer code

Description automatically generated

Now that empty values were handled, I wanted to get a sense of what my values looked like as a whole so I used .describe to give me some basic statistics (reference screenshot below). The standout values to me were ‘ANCrecovery’ being a million at max (clinically does not make sense) and the fact that antigen and allele had negative means and negative values. After inspecting the ‘ANCrecovery’ values, I can see that the million value is used as a placeholder for ‘unknown.’ Therefore, I decided to drop rows with values of ‘ANCrecovery’=1000000.

A screenshot of a calculator

Description automatically generated

The ’Antigen’ and ‘Allele’ variables refer to a count of differences between donor and recipient antigens, and the same for alleles. As you can see, the value should be a count, but the person who coded this dataset before me coded ‘No differences’ as ‘-1’, and ‘One difference’ as 0, and so on. To me, this is confusing to interpret. I felt ‘Antigen’ and ‘Allele’ could be argued to be categorical or numerical as presented in this dataset, since the values do represent distinct ordinal categories, but in theory they can also be a count of antigens or alleles (though it doesn’t seem that way because of the ‘-1’ value).

Therefore, I decided to err on the side of caution and re-encode these variables using Ordinal Encoding to handle the ‘-1’ label. I imagine it might not make much difference to my model because they might still be interpreted as counts by the model, but in terms of human understanding, I thought it would be much easier to analyze and interpret results if I could interpret each numeric value in these columns as directly being the actual count of differences.

I will note that one reference mentions it is sometimes recommended that predictors are centered to have a mean of 0, which might be what the previous user was aiming for by making ‘no difference’ have a value of ‘-1’ and ‘one difference’ to have a value of 0. However, the same reference says that whether you center or scale variables for regression models, estimates are adjusted and p-values will be the same, so I will continue with re-encoding the data for ease of human reading and understanding and to enforce ordinal rank (Mathieu\_r, 2012).

I used Ordinal Encoding from the scikit-learn library to accomplish this. As seen below, -1 was encoded to 0, 0 was encoded to 1 (not shown), 1 was encoded to 2, etcetera.

Before Ordinal Encoding:

A screenshot of a graph

Description automatically generated

After ordinal encoding:

A screenshot of a graph

Description automatically generated

Next, since machine learning data has to be numerical, I needed to handle the ‘Disease’ variable, which has string values. I used OneHotEncoder from scikit-learn library to encode the disease diagnoses to a pattern of binary values (Nolan, 2023). A ‘1.0’ value under the diagnosis would indicate which diagnosis was represented.

Now that I have dealt with all the missing values, non-sensical values, and categorical values, I wanted to get a further sense of the importance of my features (Malato, 2021). Random Forest is an algorithm that can be used to calculate feature importance, so I used it to show which features contributed most to ‘ANCrecovery,’ my dependent variable (graph seen below). I could have used this method to help with feature selection of my original dataset of 37 variables, but since domain knowledge helped to reduce those already, I am using Random Forest just to give me a better sense of feature importance among the variables that remain. My code is below.

A screenshot of a computer

Description automatically generated

A graph with blue squares

Description automatically generated with medium confidence

The results were basically as expected. The independent variables I had first chosen to keep from the 37 original variables were ‘ABOmatch’, ‘Antigen’, ‘Allele’, and disease diagnosis. After talking to my manager, she had suggested that I add source of stem cell and the cell dose of transplanted cells (‘CD34kgx10d6’) because those two values should make a clinical difference on how fast a patient recovers ANC.

As seen on the graph above, ‘CD34kgx10d6’ does have a comparatively strong effect over the other features. Based on the consultation with my manager, if stem cell source should have a clinical effect on ANC recovery, then I am happy to see that the other variables I kept (‘ABOmatch’, ‘Antigen’, and ‘Allele’) have comparatively higher scores than ‘Stemcellsource.’

Since disease diagnoses showed the lowest importance scores, I wanted to further explore the ‘Disease’ variable. To show how Disease diagnosis relates to ANC recovery, I used a bar plot and a box plot and calculated some basic statistics of average, median, min, max, and range (see below). Upon visual inspection, ‘ANCrecovery’ looked similar across diseases.

A graph of a number of blue rectangular bars

Description automatically generated

A diagram of a number of blue squares

Description automatically generated

A table with numbers and letters

Description automatically generated

Since the differences seemed so minimal, I felt the need to do an analysis of variance (ANOVA), which would determine if the differences in means between samples are significant (Carpenter, n.d.). The result was  . If p-value is below 0.05, the results are statistically significant, i.e., unlikely to be due to chance. My pvalue was 0.3, which is greater than 0.05, so the differences seen across diseases are determined to be statistically insignificant. Therefore, I decided to drop the ‘Disease’ feature. My final dataset for analysis then looked like the following, with ‘ANCrecovery’ being my target or dependent variable and all other variables being the independent variables:

A close up of a screen

Description automatically generated

Exploratory Data Analysis and Descriptive Methods

I used descriptive methods first to get an idea of what my variables looked like. I

created a histogram of ANC recovery days to see the distribution; see graph below. It appears the data does skew to the right. This might mean some higher values will pull the mean more upward than the median. The graph below shows that while most people in this dataset have lower recovery days (since the bulk of data is to the left), there are a few cases with much higher recovery days. I also used pandas library to calculate skew, with result of “Skewness: 0.7817962393762705.” The positive value indeed confirms a right skew. Since I do not have a normal distribution, which is one of the assumptions of linear regression (Statistics Solutions, n.d.), my model might not be very accurate.

A graph with blue squares

Description automatically generated

I then used the seaborn library to create a pairplot in order to get a better sense of all the relationships between variables. A pairplot is a collection of scatterplots (or histograms for when the variable is paired to itself).

A screenshot of a graph

Description automatically generated

I will “zoom in” on the last row, excluding the ‘ANC recovery’ versus ‘ANC recovery’ histogram I just discussed.

A graph of a number of blue dots

Description automatically generated

For ‘Stemcellsource’ (bone marrow source=0 , peripheral blood source=1) versus ‘ANCrecovery’, the peripheral blood group seemed to have a higher concentration of dots gathered near the lower ‘ANCrecovery’ area, around ANCrecovery=14 to 15. The bone marrow group seemed to have the darkest concentration of dots nearer to ANCrecovery=17. The peripheral blood group thus seemed to have lower ANCrecovery days than the bone marrow group, which is actually clinically accurate.

The ‘CD34kgx10d6’ versus ‘ANCrecovery’ seems to have a very loosely negative linear relationship (as CD34 dose increases, ANC recovery days decreases), which actually does make clinical sense as well.

The ‘ABOmatch’ (matched blood type=0, mismatched=1) versus ‘ANCrecovery’ plot doesn’t seem to have the most defined relationships visually. However, one relationship that can be seen here is that the mismatched group’s *range* of data points are generally higher than the matched group’s *range* of data points. (Of note, the description of the data variables seen in the metadata provided by UCI has a typo and mistakenly says both matched and mismatched blood types = 1. However, on examination of the original dataset’s values of ‘DonorABO’, ‘RecipientABO’, and ‘RecipientRh’, I can deduce that 0 indicates matched blood types and 1 indicates mismatched blood types between donor and recipient.

Similar to ‘ABOmatch’, the graphs for ‘Antigen’ and ‘Allele’ do not seem to show strong relationships. However, one thing to note is that for both ‘Antigen’ and ‘Allele’ graphs, there was more weight in the ‘0 differences’ group at lower ‘ANCrecovery’ values than the 1, 2, 3, or 4 groups. As a reminder, ‘Antigen’ and ‘Allele’ features are a count of how many mismatches there are between donor and recipient for antigens and alleles.

In summary, on visual analysis, there are some very general relationships between the dependent variables and ANC recovery. It is difficult to determine visual linear relationships for the features with binary values, but there does appear to be a weak negative linear relationship between the continuous variable CD34 cell dosage and my continuous dependent variable of ANC recovery.

Predictive Method

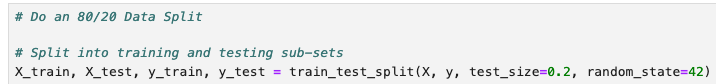
My predictive method is a linear regression model. More specifically, my model is a multiple regression model because there is more than one independent variable (Jones, 2004). Independent variables for this model are 'Stemcellsource', 'CD34kgx10d6', 'ABOmatch', 'Antigen', and 'Allele'. My dependent variable being predicted is ‘ANCrecovery’.

The first step was to define my dependent and independent variables; code shown below.

A white rectangular object with text

Description automatically generated with medium confidence

Next, I needed to split my data into 80% training data and 20% testing data; code shown below.



Next, I defined my linear regression model as ‘model’ and then using .fit trained the model on the training data that I provided (X\_train, y\_train).

A screenshot of a computer

Description automatically generated

Then, using .predict I applied the trained model to make predictions on the data that I provided (In this case, I gave the model the X\_test data from the data split).

A close up of a sign

Description automatically generated

Assessment of the Hypotheses for Acceptance or Rejection

Below, I have provided a modified version of the hypothesis table I created in Task A for my null (H0) and alternate (H1) hypotheses. To be able to accept or reject my main hypothesis, I need to determine if each independent variable had a significant effect on ‘ANCrecovery.’ This can be done by finding p-values for each variable. Generally, a p-value less than 0.05 means there is significant evidence to reject the null hypothesis (Sampaio, 2023).

In a linear regression model, the residual is the difference between the actual value and the predicted value. Ordinary Least Squares (OLS) is a way to make the line of “best fit” by minimizing the sum of the squared residuals (Sampaio, 2023). After I apply OLS on the data, a summary table is output which has p-values listed under the label P>|t| (Torres, 2023); see screenshots below.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **VARIABLE NAME** | **NULL HYPOTHESIS H0** | **ALTERNATE HYPOTHESIS H1** | **p-value** | **REJECT H0?** |
| **MAIN HYPOTHESIS** | **None of the dependent variables below have a significant effect on ANC recovery** | **At least one of the dependent variables has a significant effect on ANC recovery** | **n/a** | **Yes, reject H0 and accept H1 because at least one of the variables did have a significant effect (stem cell source and CD34 dose)** |
| “Stemcellsource” | No significant effect on days to ANC recovery between stem cell sources | Stem cell source has a significant effect on ANC recovery days | 0.000 | Yes, reject H0. Accept H1. |
| “CD34kgx10d6” | Cell dosage has no significant effect on ANC recovery days | Cell dosage has a significant effect on ANC recovery days | 0.004 | Yes, reject H0. Accept H1. |
| “ABOmatch” | Blood type compatibility has no significant effect on ANC recovery days | Blood type compatibility has a significant effect on ANC recovery days | 0.695 | Cannot reject H0 |
| “Antigen” | Number of antigen mismatches has no significant effect on ANC recovery days | Number of antigen mismatches has a significant effect on ANC recovery days | 0.790 | Cannot reject H0 |
| “Allele” | Number of allele mismatches has no significant effect on ANC recovery days | Number of allele mismatches has a significant effect on ANC recovery days | 0.144 | Cannot reject H0 |

In summary, I can reject my main null hypothesis because at least one of the dependent variables (CD43 and stem source source) was shown to have a significant effect by the p-values; this invalidates the null condition. Rejecting the null hypothesis means that I can **accept** **the alternate hypothesis**, which is that there is a significant effect between at least one of the dependent variables and day to ANC recovery.

Assessment of Model’s Accuracy

The “only way to determine [a linear regression model’s] accuracy is through residuals, which are the difference between actual and predicted values (Ogunbiyi, 2022). Various metrics make use of this concept to evaluate a linear regression’s performance. These metrics include R-squared (R2), Mean Absolute Error, Mean Squared Error, and Root Mean Squared Error (Brownlee, 2021).

R2 (or R-squared) ranges from 0 to 1 and determines the model’s accuracy in terms of the residual. Scikit-learn has a function to calculate R2. My calculated R2 was 0.143, which means that my model explains 14.3% of the variance in my target variable (Torres, 2023). One reference calls R2 the accuracy of a model (Ogubiyi, 2022), so 14.3% is quite low accuracy; see screenshot below for details:

A screenshot of a computer code

Description automatically generated

Mean absolute error (MAE) is an average of all the residuals. Basically, it is an average error, and the unit of the MAE is the same as the dependent variable. Scikit-learn has a function to calculate MAE (Nantasenamat, 2020). My MAE was just under 2 days. If I consider that the highest ‘ANCrecovery’ in my dataset is 26 days, then 2 days seems to be quite a big error. See screenshot above for my code.

Mean squared error (MSE) is the average of the squared residuals. The squaring handles any negative values and also has the effect of punishing big errors, thus magnifying their impact in the MSE. My MSE was 8.4 days squared; the code is above. The unit of days-squared is a little difficult to interpret so how do I interpret my MSE? Actually, this is where root mean squared error comes in.

Root mean squared error (RMSE) is just the square root of the MSE and returns a unit that is more easily interpreted. My RMSE then will give me back a unit of days, which can now be easily interpreted. My RMSE was 2.9 days, which seems to be a big error if I consider that the highest ‘ANCrecovery’ value in my dataset is only 26.

Based on my results for the four metrics above, the accuracy of my model seems low thus far.

I also wanted to visually assess my residuals by plotting them in a histogram; see below. Ideally, the residuals should be normally distributed. Unfortunately, my graph has a right skew. Since it is not normally distributed, this means that some statistics, like p-values, might be inaccurate.

A diagram of a distribution of residuals

Description automatically generated

I also wanted to use a scatterplot to show actual y values versus the predicted y values; ideally, both of those datasets would be identical and then have a slope of 1 when graphed against each other. The two scatterplots below actually show the same data, but the second plot additionally shows the line at which the predicted value is exactly equal to the actual value. So, ideally, these data points should fall along the green line. In my case, it was seen that there were a number of values relatively far from the green line, visually showing that the accuracy of the model is likely low.

A graph with blue dots

Description automatically generated

A green line with black dots

Description automatically generated

At this point, I referred to the OLS summary again to look at my Adjusted R-squared. Unlike the usual R-squared score, the Adjusted R-Squared adjusts the R2 appropriately as the number of predictors increase (Jones, 2004); therefore it is more suitable for my model which has various independent variables. My adjusted R-squared is 0.267, which indicates that about 26.7% of the variance of y can be explained by my model. I will show the OLS summary again from before for easy reference:

A screenshot of a computer

Description automatically generated

One note I wanted to make is that the R-squared shown in the OLS Regression Results is 0.292, while earlier it was calculated with scikit-learn’s r2\_score function as 0.143. This is because when I used the r2\_score function, it was calculated on the test data. The OLS model that provided the summary results above was given training data, so its R-squared is a little different. Regardless, evaluating both splits of data sets still shows that my model’s scores for R-squared of my training data, R-squared of my test data, and Adjusted R-squared are still poor. My model’s accuracy continues to show as being somewhat low.

Revisions

Since my accuracy was proving to be low, I wanted to try to improve my results by dropping everything but the statistically significant variables. I dropped the variables ‘ABOmatch’, ‘Antigen’, and ‘Allele,’ which were variables that the OLS Summary Results showed to have big p-values and were not statistically significant on the ‘y’ (the ‘ANCrecovery’). Code shown below.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

When I ran OLS again on my data, both my R-squared and Adjusted R-squared scores actually decreased a little bit and got worse, as seen in the screenshot below. Before dropping what I had considered to be my poor predictors, my R-squared had been 0.292 and Adjusted R-squared had been 0.267.

A screenshot of a computer

Description automatically generated

Additionally, all the error metrics for the revised model proved to be a little worse as well. See below.

Error metrics before revision:

A screenshot of a computer error

Description automatically generated

Error metrics after revision:

A screenshot of a computer error

Description automatically generated

I also re-ran the plots of predicted values versus actual values, which also visually confirmed the worse performance of the revised data and model. As seen below, the dots seemed to flatten out just the slightest bit closer to a slope of zero than the previous plots did.

Before revision; results shown below:

A graph with blue dots

Description automatically generatedA green line with black dots

Description automatically generated

After revision; results shown below:

A graph with blue dots

Description automatically generated A green line with black dots

Description automatically generated

After seeing the worse performance, for my final data product I decided to revert back to using my dataframe from before the revision.

In general, there are some assumptions that must be met for a linear regression model to be significant and efficient (Torres, 2023). These include: (1) Linear relationship between independent and dependent variable, (2) little to no multicollinearity, (3) observations should be independent of each other, (4) residuals should be normally distributed, and (5) variance of residuals should be constant across all levels of the dependent variable.

For the first assumption, my pairplot from my Exploratory Data Anlysis section showed that there was a weak linear relationship between CD34 dose and the ANC recovery days. All the other variables did not really show a linear relationship, so this likely affected my model’s fit to the data.

As for multicollinearity, a few of my variables could be considered multicollinear. For instance, ‘Antigen’ and ‘Allele’ are usually related clinically as they both relate to similarity of genetic and immunologic material. Usually, when one of these moves in one direction, the other does too. The correlations between my independent variables are shown visually on the heat map below. It is seen that ‘Antigen’ and ‘Allele’ do indeed have the highest correlation with 0.68. We can also calculate Variation Inflation Factor (VIF) to find the degree of correlation; VIF of 1 is “not correlated,” 1 to 5 is “moderately correlated,” and >5 is “highly correlated” according to Paul (2021). My code below shows that all my VIF’s are between 2 and 4. Since VIF is “the degree that the variances in the regression estimates are increased due to multicollinearity” (Statistics Solutions, n.d.), it appears my model has moderate multicollinearity, which likely decreases my model’s performance.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

For the third assumption (that observations be independent), my variables are somewhat related to each other because they are all related to one patient’s disease profile. Since the variables all have very complex medical or physiological relationships to each other within the human body, their subtle influences on each other are beyond my understanding and is limited by my education and training. All that being said, my point is that their influences on each other might not be captured well by a simple regression model, so when the model tries to determine how much an independent variable actually contributes to change in ‘y’ the results might be misleading. Thus, it is likely that my data does not meet the third assumption to some degree.

Residuals should be normally distributed, but my plot (shown previously) demonstrates a right skew of the residuals. Thus, my linear regression model’s performance was also likely affected by my residuals.

For the last assumption, this is a reference to something called homoscedasticity (Torres, 2023), which can be shown visually with a scatterplot of residuals versus predicted values. Ideally the plot should have no pattern. If it looks cone-shaped, it is heteroscedastic and a non-linear type of analysis might be more appropriate (Statistics Solutions, n.d.). Based on my plot below, it is a little hard to tell, but I do see somewhat of a cone with the wider end to the right, so I can see that possibly a linear regression model might not have been the best choice for my data.

A screenshot of a computer program

Description automatically generated

A diagram of a graph

Description automatically generated with medium confidence

Final Thoughts

In summary, metrics (like adjusted R-squared and MAE) and graphs (like my scatterplot of predicted y’s versus actual y’s) have shown that my model’s accuracy is low. Additionally, when I tested for how well my model met linear regression assumptions, we can see that my dataset did not quite meet the assumptions very well. On starting this project, I had expected my model’s performance to be poor due to the small dataset of only about 180 records, and it seems my initial concern was valid. My small dataset had a right skew, but linear regression models work best with normally distributed data, so from the start, it seems my initial dataset was not ideal.

In response to the poor error metrics, I had hoped to improve accuracy of my model by reducing number of features (by dropping the features found to have no statistical significance). However, since my model essentially got a tiny bit worse after revision, for my final data product I decided to revert back to my dataframe from before the revision.

Given the chance to do this project over again, I would try to address each of the assumptions of the linear regression model, whether that means I choose different features or transform the data differently. Or, possibly, I might have to try other machine learning models to see if they are a better fit.

Regardless of poor accuracy, when I test my product with my user interface, it generally seems to give a prediction that at least increases or decreases in the expected direction. In other words, the prediction seems to move in a direction that makes clinical sense when I change certain patient features, so I am satisfied with this performance for my first machine learning project.

# Quick Start Guide

I have uploaded my Jupyter Notebook file to Google Colab. I will share the link with the project evaluator.

<https://colab.research.google.com/drive/1FocwHEEZs3rYAgSegSW7ekBBaxxUtoQy?usp=drive_link>

**Step 1**. Please click on the given link. This should open Google Colab. The notebook is titled final\_project\_notebook.ipynb

A screenshot of a computer

Description automatically generated

**Step 2**. Please click on “Runtime in the menu. Then click “Run All.” This should run every cell.

A screenshot of a computer

Description automatically generated

**Step 3**. Scroll all the way to the bottom of the notebook. You will see the user interface that looks like the screenshot below.

A screenshot of a computer screen

Description automatically generated

**Step 4**. To use the product, please select the desired patient characteristics or type in as necessary.

**Step 5**. Click the “Predict” button to see a prediction by the machine learning model.

**Step 6**. If you desire, the raw .csv file containing the dataset for this project is also available at the link below.

<https://drive.google.com/file/d/1U2NnX2Uvuw0YQ1E93IrXYwirLuc8dl6t/view?usp=drive_link>

# References List

AABB, America’s Blood Centers, American Society for Blood and Marrow Transplantation, Cancer Transplant Institute, Center for International Blood and Marrow Transplant Research (CIBMTR), Children’s Mercy Hospital, CORD:USE Cord Blood Bank, Dana-Farber Cancer Institute and Brigham and Women’s Hospital, Froedtert & The Medical College of Wisconsin, H. Lee Moffitt Cancer Center, Michigan Blood, National Marrow Donor Program/ Be The Match, New York Blood Center, Ochsner Medical Center, Penn State Health Milton S. Hershey Medical Center, Scripps Health, Spectrum Health Cancer Center, Stanford Health Care, Texas Transplant Institute, … Wake Forest Baptist Comprehensive Cancer Center. (2018). *Letter to Seema Verma*. Letter. <https://www.aabb.org/docs/default-source/default-document-library/positions/comments180614.pdf?sfvrsn=6e0ef495_10>

Brownlee, J. (2021). *Regression metrics for machine learning*. Machine Learning Mastery. <https://machinelearningmastery.com/regression-metrics-for-machine-learning/>

Carpenter, A. (n.d.). *What is ANOVA (Analysis of variance) testing?*. Qualtrics. <https://www.qualtrics.com/experience-management/research/anova/>

GeeksForGeeks. (2024). *Feature importance with random forests*. GeeksForGeeks. <https://www.geeksforgeeks.org/feature-importance-with-random-forests/>

Institute of Data. (2024). *What are the 5 steps in data science lifecycle*. Institute of Data. <https://www.institutedata.com/us/blog/5-steps-in-data-science-lifecycle/>

JMP. (n.d.). *Interpreting results in explanatory modeling*. JMP Statistical Discovery. <https://www.jmp.com/en_us/statistics-knowledge-portal/what-is-multiple-regression/mlr-residual-analysis-and-outliers.html>

Jones, J. (2004). *Multiple regression*. James Jones, Professor of Mathematics. <https://people.richland.edu/james/ictcm/2004/multiple.html>

Malato, G. (2021). *Feature selection with random forest*. Your Data Teacher. <https://www.yourdatateacher.com/2021/10/11/feature-selection-with-random-forest/>

Mathieu\_r (2012, June 4). *When conducting multiple regression, when should you center your predictor variables & when should you standardize them?* [Online forum post]*.* StackExchange. <https://stats.stackexchange.com/questions/29781/when-conducting-multiple-regression-when-should-you-center-your-predictor-varia>

Nantasenamat, Chanin. [Data Professor]. (2020, April 1). *Machine learning in Python: Building a linear regression model* [Video]. YouTube. <https://youtu.be/R15LjD8aCzc?si=phhRVjhYmm_g8Htm>

Nolan, Ryan. [Ryan Nolan Data]. (2023, August 14). *One hot encoder with Python machine learning (Scikit-learn)* [Video]. YouTube. <https://www.youtube.com/watch?v=rsyrZnZ8J2o&list=LL>

Ogunbiyi, I. A. (2022). *Top evaluation metrics for regression problems in machine learning*. FreeCodeCamp. <https://www.freecodecamp.org/news/evaluation-metrics-for-regression-problems-machine-learning/>

Paul, Sanchita. (2021). *How to check for assumptions in a linear regression*. Medium. <https://medium.com/analytics-vidhya/how-to-check-for-assumptions-in-a-linear-regression-a68116aef88a>

Roepke, Brian. (2024). *4 methods that power feature selection in a machine learning model*. Data Knows All. <https://dataknowsall.com/blog/featureselection.html>

Sampaio, V. (2023, June 2). *Understanding ordinary least squares (OLS): The foundation of linear regression*. <https://medium.com/@VitorCSampaio/understanding-ordinary-least-squares-ols-the-foundation-of-linear-regression-1d79bfc3ca35>

Sanjyal, Ankit. (2022, December 24). *Dimensionality reduction vs feature selection*. Medium. <https://medium.com/@asanjyal81/dimensionality-reduction-vs-feature-selection-e68f91aa8724>

Stanford Medicine. (n.d.). *Bone marrow transplant for children*. Stanford Medicine Children’s Health. https://www.stanfordchildrens.org/en/topic/default?id=bone-marrow-transplant-for-children-90-P03062

Statistics Solutions. (n.d.). *Assumptions of multiple linear regression*. Complete Dissertation by Statistics Solutions. <https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/assumptions-of-multiple-linear-regression/>

Team Asana. (2024). *Business requirements document template: 7 key components, with examples*. Asana. <https://asana.com/resources/business-requirements-document-template>

Torres, Luís Fernando (2023). *Mastering linear regression with statsmodels*. Medium. <https://medium.com/latinxinai/mastering-linear-regression-with-statsmodels-95233a2a602e>