**Individual Portfolio, Data Challenges 4**

**Week 2:**

This week, our groups were assigned, and we got an insight into our project allocations. I was assigned to my first preference, which is the Catheter line positioning project. With the guidance of Simon Clarke, our course coordinator, I gained valuable insights that have set the tone for my journey in this project.

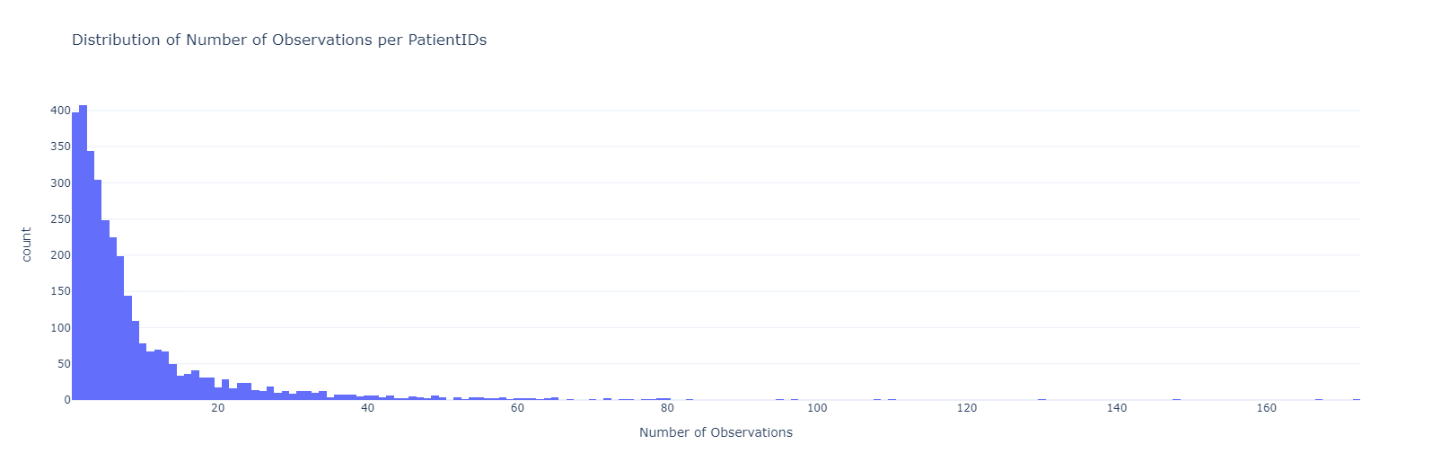
Simon introduced us to various key points that will undoubtedly shape our project. One critical aspect he highlighted was the potential necessity of achieving balance in catheter positions within the dataset. Moreover, he emphasized the significance of segmenting our dataset into distinct categories for training, testing, and validation. The validation phase looks like an important step in all the projects that involve image classification.

While I don't yet possess an in-depth understanding of several concepts he mentioned, such as saliency maps, backpropagation algorithms, convolution, and neural network architectures like DenseNet, ResNet, and U-Net, I would have to go through a self-learning journey to understand these concepts thoroughly.

I am excited about working on this challenge. I know it is not an easy one, but it is a great opportunity to master image classification concepts.

**Week 3:**

This week, I have dived into data wrangling and exploratory data analysis. Interestingly, the catheter dataset stood out with its impeccable quality - no missing values or duplicate records required any attention.

After assuring the data's cleanliness, I plotted the features of the data to gain an understanding of the data. One of the plots I created was this one showing the number of observations per patient. Based on that, the data has around 30000 observations, but only 3000 patients. In other words, many patients had catheter X-rays taken more than once.

Next, I plotted the number of observations for each catheter and its position; the majority of the catheters are CVC catheters in all three positions: abnormal, borderline, and normal.A blue and white rectangle

Description automatically generated

Upon seeing that plot, I realized that the data might be unbalanced concerning positions, which was mentioned by Simon last week, so I made another plot that shows how many catheters are in each position, and it appears that most are placed in the normal position. Percentage-wise, 70.92% of catheters are placed in the normal position, 21.53 % in the borderline position, and 7.55% in the abnormal position.

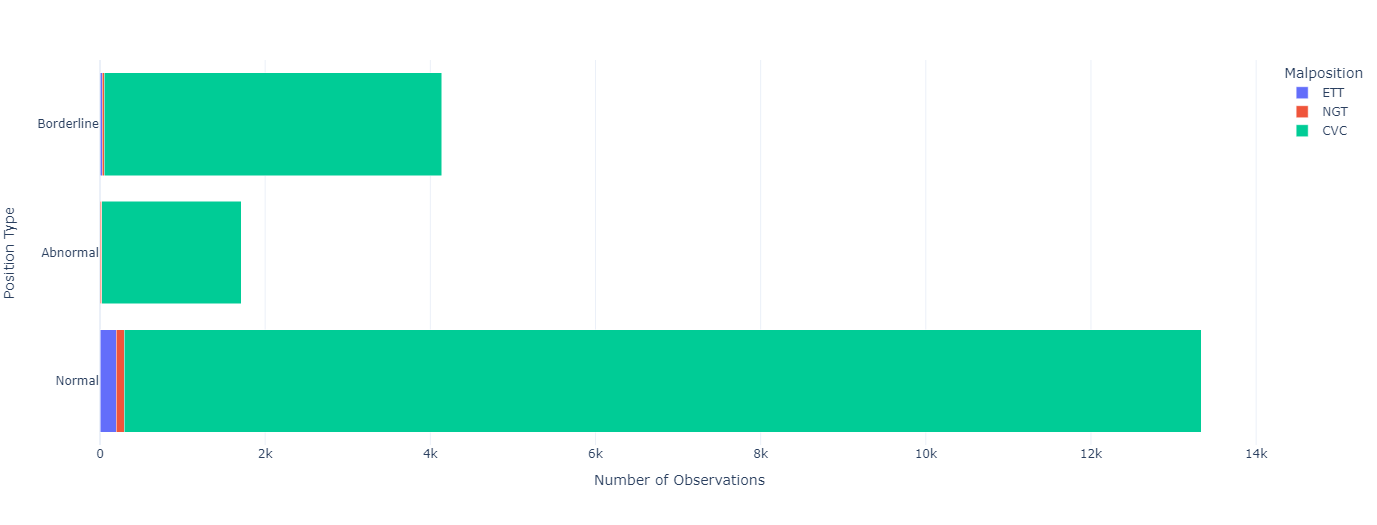
A green and orange bar graph

Description automatically generated

Therefore, the data is highly unbalanced. Hence, my focus next week will be on balancing this data set.

**Week 4:**

This week I started working on balancing data. There are three different ways to balance the data, over-sampling, under-sampling, and SMOTE. As a starting point, I have tried both over-sampling and under-sampling, as SMOTE requires independent and dependent variables, X and Y, and I still don't have an independent variable, which would be the NumPy arrays or the pixels arrays. Since all the categories are highly imbalanced, I found it challenging to try to balance all nine categories. Therefore, I was wondering if there was a reasonable way to deal with a smaller data set, such as dropping observations. So, I attempted data processing again, and I discovered that some images were classified in different categories, like ETT – Normal, ETT – Abnormal, and CVC – Normal, which was strange. The internet says that people can have multiple catheters at the same time, but I was unsure about having two catheters of the same type in different positions, so I dropped the observations that classified the X-ray in more than one category. As a result, I ended up dropping 10,000 observations, and since my laptop is not a supercomputer, I found dropping those would speed up the model. This is how the dataset looked like after dropping them.



Having dropped those observations, I ended up with almost all CVC catheters, and after discussions with my colleagues, we decided to drop the other catheters and focus on classifying the positions for the CVC catheter for now. It is clear that the data is still imbalanced, so I just under-sampled the Normal and Borderline positions to match the number of observations in the Abnormal positions, this is just for now.

**Week 5:**

This week I focused on transforming the images into arrays of pixels because apparently, this is the way to do image classifications. I tried different functions from different libraries. One library I used was matplotlib, which has the "imread" function that reads images as pixels, and I used the "resize" function to ensure all arrays were equally sized. I did that on the under-sampled data from last week, and it took about an hour and a half to process all of them, which is insufficient. Consequently, I searched for other libraries, and I found a library called Pillow, an updated version of Python Image Library (PIL). In order to read the images as arrays of pixels and resize them, it took 28 minutes, about a third of the time that took matplotlib library functions. As soon as I had completed this essential step, I merged it with the training data set that has the images classified into the catheters and their positions, so, now this is how the final data set looks like.

A screenshot of a computer

Description automatically generated

Then I tried saving the data frame in CSV format, but whenever I tried to save it, it was saving the resized pixel arrays as a string by adding an extra "/n" between the arrays, preventing me from viewing the images. I tried different ways to get rid of them, but despite my efforts, I couldn't fully resolve the issue of CSV files containing unwanted "/n" characters between the pixel arrays, so for the time being I would have to run the pixeling function each time I want to work on the data. This is so I can move forward with data modelling.

**Week 6:**

This week, my group and I came across an issue regarding the data type of the resized pixel arrays. It seemed that the data type was categorized as an "object" when checking within the Data Frame, which we thought might be an issue when we were going to build the neural network because neural networks expect the pixel arrays in a data type called NumPy arrays. We thought that it was showing an object because it was saving it as a string or something, but interestingly, this "object" classification in Python doesn't necessarily imply a string type; it simply represents the data type in an abstract way within a data frame. So, after a couple of hours trying to figure out the problem, it turned out that there was no problem in the first place.

Now that we have resolved almost all of our image preprocessing concerns, I will now finally move forward to image classification. One of the most popular modelling techniques that I found for image classifications is neural networks. I will focus now on creating, training, and evaluating an initial neural network model. I will try to proceed to craft an initial neural network using Keras, a remarkable high-level API compatible with TensorFlow.

**Reflecting on the Group Project Experience:**

Through this experience so far, I've mainly gained technical insights and valuable lessons about effective teamwork and problem-solving. Discussing different points of view and different opinions and finding common ground was always required. This experience emphasized the significance of establishing clear roles and responsibilities and creating an environment where each team member's strengths could be effectively utilized to achieve our goals.

One significant hurdle was that everyone was so busy with the other units and other assessments, so we did not have a lot of meetings outside the class, which slowed our progress. We overcame it by deciding that we would have a weekly meeting every Thursday to see where everyone was up to and to make sure that we had something to present in the weekly standup.

My primary contribution to the project was focused on data preprocessing and model architecture design. By delving deep into these aspects, I aimed to create a strong foundation for our project. Additionally, I actively participated in brainstorming ideas during the workshops.

While reflecting on my performance, I would say that it is related to the hurdle that we faced, like timely task completion and regular progress updates were essential, and I could have been more proactive in ensuring consistent communication. In the rest of the semester, and in future projects in general, I intend to refine my time management skills, allocate time for unforeseen challenges, and maintain more consistent interactions with my team members.

**Week 7**

This week, while my original plan was to start with the modelling and neural networks, I had to adjust my focus due to our coming mock presentation scheduled for next week.

Our team came together for a productive meeting, during which we outlined each member's designated speaking points. After that, I got to work on the slides. I started with our research questions and then moved on to outline our project pipeline, which starts with the exploratory data analysis (EDA), then understanding how to read images effectively, and dealing with the essential task of data balancing. At this point, our models weren't quite ready yet, so our presentation mainly revolved around these foundational aspects of our project.

Our presentation itself went smoothly, but we received some valuable feedback from Zach. He pointed out that we needed to rework our research questions, so we will make sure to rephrase and refine them as we are working on the modelling as well.

**Week 8**

This week I was learning neural networks in depth for the first time. This topic is so complicated yet so interesting. I learned how to create my own sequential model and define my own layers, so I implemented one for the project. My model's architecture included an input layer, a sequence of dense layers, dropout layers for regularization and model fine-tuning, and lastly an output layer with three classes. This multi-class setup was essential since I am trying to create a model that predicts three different positions for the CVC catheter.

Selecting the right optimizer and learning rate was a hard task. I experimented with different optimizers, such as Adam and SGD, while varying the learning rates. I was amazed at how these choices impacted the accuracy of my model. So, I was not sure which one to choose, so, I started looking for an approach similar to grid search, which we often use for regression models to find the best hyperparameters, I found a technique called random search. This technique involves exploring various hyperparameters through multiple trials, and I get to select the number of trials, and then it pinpoints the combination that yields the best results.

A screen shot of a computer program

Description automatically generated

After I was done with selecting the best hyperparameters, I realized that I needed to normalize the pixels before fitting the model, I achieved this by employing libraries like OpenCV (CV2) and the Python Imaging Library (PIL) to standardize pixel data. Following this, I divided my dataset into training, testing, and validation sets, setting the stage for model training.

Upon training the model with the optimized hyperparameters, I achieved a respectable accuracy of 0.5. While this result is promising, I acknowledge that there's room for further improvements. I'm currently contemplating potential adjustments, whether it's enhancing the model architecture by introducing additional layers, refining the dataset by increasing the sample size or altering its dimensions. These considerations will be discussed further during my upcoming meeting with my project group members as we collaborate to fine-tune our approach.

**Week 9**

During this week's workshop, our course coordinator, Simon Clark, delved into the topic of regression neural networks. He also shed light on the concept of callbacks that can be integrated into our models. One such callback is early stopping, which caught my attention previously while I was implementing my own model. However, Simon's explanation clarified its purpose and functionality, making it much easier for me to grasp. I took the opportunity to share my current model with him and received valuable feedback in return. Simon also suggested considering utilising built-in models like ResNet-50 and DenseNet-121, which I plan to explore since the accuracy of my current model isn't quite where I would like it to be.

I was also working with Colin, one of my group members, on learning how to use MASSIVE, which is like a supercomputer boasting higher RAM capacity than Google Collab, luckily, we have access to it since we are working on the project. However, we encountered a couple of problems, particularly concerning directory management and missing libraries on MASSIVE. For example, Plotly, which is one of the libraries that we rely on for exploratory data analysis (EDA), required installation and definition. We successfully addressed this issue. However, we were not able to install and define CV2, which we are using for normalizing pixel data, so I will look for alternative libraries with similar functionality to CV2 and attempt their implementation instead.

Looking ahead to the next week, my primary goal is to set up and experiment with ResNet-50 and DenseNet-121 to assess whether they yield improved accuracy compared to my custom model. And now that I know how to use MASSIVE, I should be able to get the work done way faster.

Week 10

During this week, my primary focus was on adapting the data to fit ResNet-50 and DenseNet-121, both of which are pre-built models. These models come with numerous predefined layers, eliminating the need to manually specify all the layers. Instead, you only need to define the input and output layers, make use of some of the pre-defined layers, and potentially freeze others. It is not typical to employ all the layers since this would result in a highly complex model that takes a long time to run. Additionally, you have the flexibility to incorporate additional layers as needed.

ResNet-50, as the name suggests, comprises 50 layers, while DenseNet-121 consists of 121 layers. I conducted experiments by adjusting the architecture and various hyperparameters of both models. However, the accuracy I achieved was almost identical for both, hovering around 0.39.

In response, I decided to modify the output layer, reducing the prediction classes to just two: "abnormal" and "normal." This entailed excluding observations related to borderline catheters. This adjustment yielded an accuracy of 0.55 for both models.

Despite the similarity in results, I ultimately decided to proceed with ResNet-50 due to its greater popularity. Nevertheless, I intend to thoroughly explore the architecture and inner workings of both models to gain a deeper understanding of their characteristics.

One thing I want to try next week is to use the annotation data which would probably increase the accuracy of the models.

Week 11

This week, I embarked on an attempt to enhance the accuracy of our results using annotation data. My approach involved overlaying the annotations onto the original images, saving these modified images as new ones, and then converting them into pixel data using the PIL library. Subsequently, I employed these pixel data in our machine-learning models. The outcomes of this effort were as follows: an accuracy of 0.52 when predicting three positions and an impressive accuracy of 0.78 when predicting only two positions.

Notably, during discussions with Simon Clark last week, it was suggested that focusing on predicting only two positions aligns better with our current knowledge and objectives. Following this advice, I saved the model designed for two-position predictions and applied its weights to the original X-rays without annotations. The accuracy dropped to 0.5, which is actually lower than the accuracy achieved when predicting two positions without any annotations last time, which stood at 0.55. Thus, I decided to move on with the annotations only.

With most of the modelling strategies explored, I've shifted my focus towards preparing presentation slides since the presentation is scheduled for next week. I also intend to adapt the current models to other catheters, following the two-position prediction approach. This will be valuable content to include in our final report and presentation, ensuring that we've explored a comprehensive array of strategies.

Week 12

This week, I made significant strides as I prepared for our upcoming presentation. My primary task was to apply our current model to the annotations of different catheters, beginning with the ETT catheter. During this process, I encountered a notable challenge – a substantial data imbalance. There were merely 30 observations for abnormalities compared to over 2000 observations for normal cases. This situation meant that the under-sampling technique, which worked well with the CVC catheter, was not a viable solution here as it risked information loss.

In response to this imbalance, I delved into exploring alternative data balancing methods. I considered two approaches: random oversampling (sampling without replacement) and SMOTE (sampling with replacement). My experimentation yielded intriguing results. Random oversampling essentially created duplicates, resulting in a misleading model accuracy of 100%. This was deceptive because the model essentially saw the same data during both training and testing phases.

Conversely, SMOTE, which generated synthetic images from the available data, achieved an accuracy of 97%. While this result was promising, I remained cautious about the reliability of the model's performance since it was trained on artificial data.

Considering our upcoming presentation, we collectively decided not to incorporate this latest model. This decision was based on the need to adhere to our 10-minute time constraint.

Reflecting on the project as a whole, I found it to be both challenging and enlightening. I gained valuable insights into data-balancing techniques and neural networks. With further knowledge and experience, I am confident that similar results can be achieved without relying on synthetic images. Overall, this project has provided me with a rich learning experience.

**Reflecting on the Group Project Experience:**

Throughout this project, I gained valuable insights into the significance of collaborative brainstorming sessions and discussions in the context of accomplishing a complex project. These interactions proved essential for my understanding of neural networks and their underlying principles. I encountered various challenges, particularly in grappling with entirely new concepts such as data balancing, neural networks, optimizers, and layers, all of which were unfamiliar to me when we embarked on the project. While this initially slowed our progress, the learning experience was incredibly rewarding, as it allowed us to directly implement the concepts we acquired.

In terms of my contribution to the project, I was extensively involved in various aspects, including coding for data quality, balancing, exploratory data analysis (EDA), modelling, report writing, and the preparation of PowerPoint slides. I consider myself a committed and engaged team member who played a substantial role in shaping the project's final outcome.

Looking back, there are areas where I believe I could have improved my performance. In future projects, I would strive to enhance my time management skills to streamline our progress even further with more complex concepts like segmentation.

Overall, this reflection underscores the profound lessons I've learned from teamwork on a complex project. It highlights the importance of open communication, adaptability in the face of challenges, and the substantial impact of my contributions on the project's ultimate success. This experience has provided me with valuable insights into my strengths and areas for growth in collaborative endeavors.