Michael Luke Battle
Student No: 200725640

CSC8631 – Critical Reflection

In this project I have used the CRISP-DM methodology to produce a data analysis pipeline in such a way that that the analysis is fully reproducible and can be deployed for use with additional data. Tools such as *ggplot, rmarkdown and ProjectTemplate* have significantly aided this process and in this brief report the merits and limitations of these, along with CRISP-DM, will be discussed as I critically reflect on the project as a whole.

Following the CRISP-DM process allowed for an intuitive approach to the construction of my data analysis pipeline. The first step of this process is gaining business understanding, followed by data understanding, then undertaking data preparation, modelling, evaluation and deployment. The extent to which CRISP-DM could be applied in the business understanding stage of this process was limited due to all information on the customer in the scenario being obtained in a brief project outline. Much of this stage in the CRISP-DM process is focused on gaining a deep understanding of the business, the key persons and departments, and liaising with each of these to identify problem areas where a data mining project could prove fruitful. As none of this was applicable in this project, this section of my project was brief as I listed the benefits of learning analytics and why Newcastle University may want to apply it to their online resources. Having said this, it did provide a good justification for the analysis.

The next step in CRISP-DM is to understand the data. The initial part of the CRISP-DM process for this step requires detailing the data mining process, however this is made irrelevant in this project due to the data being given to us. CRISP-DM provided a helpful guide for the remainder of this phase, in which the data is described, and quality checked. *ProjectTemplate* was of huge use for this phase, as it allowed me to create helper functions that were loaded with the project that could quickly assess some basic characteristics of all of the data files that were saved in the project directory. The CRISP-DM methodology in this phase made sense, in that even though the files were broken per course run it seems good practice to still check them all thoroughly before compiling them to avoid potential errors that could take a while to investigate. However, due to the granular nature of the data it meant that the *explore* component in this phase would be more appropriately performed after the data has been prepared to streamline the calculations.

For the data preparation stage in the CRISP-DM process, a major driver in its construction was the requirement of reproducibility such that the analysis can be easily replicated with the same data and performed on future data. There are several assumptions that were made in the design of this. Firstly, the data file formats will remain the same as they are for this analysis, and specifically have the exact same column names. Much of the data preparation in the munge files is based on specific column names, to account for the scenario that the variables are ordered differently, and ensuring the data is still prepared correctly. Secondly, as the course run corresponding to the file is taken from the name, and this drives much of the subsequent analysis, it is critical that the filename remain in same exact format and filetype. For the video-stats file, this filename format would be “cyber-security-“, followed by the run number, then “_video-stats.csv”. I assumed this would be the same each time in order to minimise the manual interaction with the file, as this allowed me to automate picking the course run number for each file. The alternative was for the file user to input the first run of the files uploaded manually, which would then lead to more assumptions being made in the data files, such as that the files that the analysis is ran on have to be corresponding to sequential course runs. *Dyplr* was used quite heavily in the data preparation phase of my data analysis pipeline, particularly the pivot_longer() function for transforming my data into long form and the bind_rows() function for compiling individual files into a data frame without growing an object incrementally (which could end up being very computationally intensive if the data analysis pipeline was used for many files). The use of the pipe in

dplyr was also useful in keeping my code more concise and readable. *ProjectTemplate* was incredibly useful for this phase of the analysis, with the "munge" folder along with the "cache" allowing for a fast, seamless loading of the project, without the need to run-through any code. Both this and the *dplyr* package have proved invaluable in ensuring the reproducibility of this phase in my data analysis pipeline. I also found the steps for this phase in CRISP-DM to be useful, providing the roadmap for an efficient workflow by cleaning, constructing then formatting the data.

As modelling was not the focus of this project, this section in CRISP-DM was of little use and this phase mainly consists of exploration of the data. *ggplot* and *rmarkdown* were hugely effective in both the production of my plots and insertion of them into my project report. As *ggplot* enables the user to create plot objects, this means that the plots can be input into a document with minimal code, enabled by *rmarkdown's* ability to source from different files. Additionally, I found *ggplot* to be significantly easier to use than the native r plotting, allowing for plots to be built in layers. The filter() function from *dplyr* was also useful in creating plots for specific values in the data. As well as this, the stat_summary() function facilitated me being able to plot averages from across my data without the need to transform the data into smaller data frames. In calculating these averages, I assumed that the videos used in the course remained unchanged for each run. This assumption was motivated by the conclusion that the duration of the videos does not change for each run, but impacted my design decisions in that I opted to take averages of audience retention over course runs. In the context of my analysis this means that the resulting values carry more weight as they come from a larger data set but would not be appropriate if the videos had changed. In this exploration, I also assumed that variables such as "viewed_five_percent" represented the percentage of learners on the course that viewed 5% of the corresponding video. While this is not explicitly stated, I assumed this due to the count variables taking integer values, whilst these variables had decimal values to 2 d.p. all less than 100, suggesting they are percentages Much of my analysis has been structured around this assumption, and it allowed me to make precise comparisons on how audience retention varied across the videos. There is also some ambiguity with how these were calculated should a learner view a video multiple times. Would the percentage viewed be taken as an average over multiple views or only from the first for each learner? If it is taken from the initial viewing, some of our conclusions won't be entirely valid as some learners may have watched the videos fully on the second viewing.

I found the evaluation phase of CRISP-DM useful in ensuring that my evaluation is thorough. It was helpful to have a checklist on hand that documents all the many facets of a data analysis pipeline to consider upon its review. Considering things such as data pipeline development in the future, how results can be enhanced, possible failures and ranking project outcomes allowed for a comprehensive evaluation of the project.

For the deployment phase, I found CRISP-DM offered some good discussion points on deployment strategy, however as the deployment phase of this particular project is not wholly sophisticated and absent of a data mining aspect, much of the advice pertaining to this was irrelevant. With regards to the deployment of the project itself, I opted to create a brief shiny presentation in three parts to mirror the three components of my analysis that provides key plot summaries. As this shiny presentation is filled with the plots resulting from the analysis, this presentation is not only reproducible to it's current form, but would provide a useful tool for summarising the analysis with any future data loaded into the data pipeline. I found shiny highly user-friendly and intuitive in creating the layout of the presentation but did not find stylistic preferences easy to implement, with them stemming from CSS.